EL SUDOKU NOSTRE DE CADA DIA, PER CORTESIA D'AUTOCAD

Carles Nogués Mañé (àlies Joan Colom), exprofessor del Departament d'Expressió Gràfica a l'Enginyeria (Universitat Politècnica de Catalunya)

Barcelona, març de 2013

ÍNDEX:

Introducció sense vaselina

Quan algú de la Universitat tria els sudokus com a tema de recerca, com si no li disgustés ser proposat per als Premis Ig Nobel, de segur que alguna cosa no va bé. És clar que encara és pitjor que hi hagi professors amb dedicació a temps complet que, a banda de donar classe i posar-se al dia en la seva especialitat, no facin elucubracions sobre sudokus ni sobre res, tret que se'ls retribueixi a més a més. I allò veritablement greu és que sigui la pròpia Universitat (que en això sí que és un fidel reflex de la societat que l'acull) qui estimuli aquestes tendències. No es gens estrany que bona part del personal de neteja desaparegui misteriosament dies sencers (deixant, això sí, el carretó amb els estris en llocs ben visibles), seguint l'exemple d'un personal administratiu que sovint s'organitza en tandes on la compareixença del titular cal deduir-la de l'enllumenat, de l'aire condicionat i de l'ordinador en marxa (però no de la seva presència física), quan al seu torn aquest no fa sinó reproduir *mutatis mutandis* les pautes de conducta predominants entre el personal docent, estament que no té l'obligació de fitxar ni el més mínim interès a escatir si els períodes anomenats "no lectius" (12 setmanes l'any, pel cap baix) són de vacances únicament per als estudiants o per a tothom. A més, tot plegat passa amb la connivència dels sindicats, tant dels corporatius com dels que s'autoqualifiquen "de classe" però tampoc no s'immuten encobrint de fet pràctiques que malbaraten recursos producte d'un esforç fiscal que recau bàsicament sobre les classes treballadores. Havent pogut treure part del que duia al pap produint-li un cert desassossec, l'autor ja se sent més relaxat i en disposició d'anar per feina.

A tots ens ha passat algun cop, si no molts, l'experiència següent: embolicar-se en una feina, un joc o qualsevol activitat relativa a la resolució d'un problema, resolució que *a priori* semblava d'allò més fàcil però que alhora presentava prous atributs de no immediatesa com per esdevenir un repte atractiu, i dies, setmanes o mesos després trobar-s'hi encara atrapat, amb el següent dilema afegit: plantar-se i deixar-ho córrer, per haver-hi esmerçat molt més temps del previst inicialment, o seguir fins a la seva resolució, precisament per justificar que aquest temps no s'ha perdut inútilment. Això li va passar a l'autor fa més de dos anys, quan es va adonar que ja en portava mig treballant en un tema absolutament intranscendent, on havia anat a raure pensant que se'n sortiria en quatre mesos. A sobre, començava a sospitar que el camí emprès no el duria enlloc. Però com que ja havia adoptat la decisió de demanar la jubilació voluntària al cap de dos anys acadèmics i entenia que era millor esperar la nova situació per reprendre un treball de més volada que havia interromput vuit anys enrera, va decidir de continuar, pensant que el tema era prou lleuger com per suportar diferents ritmes de dedicació, alternant-lo amb les classes a l'E.T.S. d'Enginyeria de Telecomunicació (Campus Nord, de Barcelona) i amb els períodes de dos mesos i mig (primers d'any i estiu) en què els exàmens i la matrícula afavorien una permanència més continuada en la seu departamental, a l'E.T.S. d'Enginyeria Industrial (Campus Sud), i una activitat I+D més profitosa.

Del perquè 15 anys d'activitat acadèmica dissociada entre els dos campus l'havien cremat fins el punt d'esperar amb candeletes la jubilació i entretenir l'espera amb sudokus entre classe i classe, ell que mai no havia estat aficionat als mots encreuats i passatemps similars, en parlarà més endavant. De moment, vol confessar que un dia es va veure a si mateix com una de les coses que més havia detestat a la vida: convertit en el típic funcionari que mata les hores remunerades llegint el diari o dedicat a la papiroflèxia, amb l'agreujant que estava comprovant en la seva pròpia pell com allò que els sudokus tenen un cert poder addictiu no era pas cap tòpic, i que la fal·lera de fer-ne almenys un cada dia, mentre esmorçava, era tan morbosa que tenia un punt de neurosi obsessiva: amb un pare que havia passat una tercera part de la seva vida amb Parkinson i una mare que patia d'Alzheimer, s'havia entestat a fer-ne ús com si la mesura de la seva capacitat per resoldre'ls fos un test que pogués anunciar si també a ell li tocaria la loteria, a temps de pensar alguna maniobra evasiva d'eficàcia provada. Va ser l'intent de rebel·lar-se contra aquesta evidència, proseguint en la línia "Joan Colom" de tirar pel dret jugant a fer recerca aplicada o desenvolupament d'aplicacions (com us quadri més) al marge del reconeixement oficial, que el va decidir a donar un cop de timó per reorientar això dels sudokus des del vessant del consum al de la producció: n'hi havia suficient per recuperar l'autoestima i, alhora, li semblava que el tema era prou frívol com per no veure's aclaparat per dificultats que el poguessin superar.

I, tanmateix, va pecar de frívol en el segon aspecte, infravalorant la frivolitat. D'entrada, la primera intuició no és que fos falsa: simplement era impracticable, com ho seria omplir un dipòsit a culleradetes, i el primer capítol se'n fa ressò. Per això ara no pretén resumir en unes poques línies el que ha necessitat més de sis-centes pàgines (un avantatge d'anar per lliure és poder passar olímpicament dels protocols establerts) i com a *abstract* haureu de conformar-vos amb el resum d'*E-prints UPC* que segurament ja us heu trobat abans d'accedir a aquest document, perquè l'autor es limitarà a dibuixar un esbós de la trajectòria real de treball en relació als 14 capítols que veieu ordenats a l'*ÍNDEX*, assimilant-la al traçat d'una N invertida (*И*, o "I" ciríl·lica): de primer, una progressió que comença amb la relliscada 1, segueix des del 2 fins al 12 (saltant, però, els 4, 5, 6 i 7), i s'amplia fins al 14; després, un retorn al 3, i finalment la progressió definitiva fins al 14 (però incorporant-hi 4, 5, 6 i 7, i revisant tots els altres capítols). Així va ser com, començat el seu últim quadrimestre (primavera de 2009), i tenint en compte que la U.P.C. no acceptava que es jubilés l'u d'octubre per completar 31 anys de servei (malgrat que el primer contracte havia estat un u d'octubre, adduïa que amb la posta en marxa dels plans d'estudis del 92, d'estructura quadrimestral, els cursos es van avançar a l'u de setembre, i que, per gaudir dels incentius de la jubilació voluntària, almenys calia treballar un quadrimestre sencer) i que ho va haver de fer l'u de setembre, quedant-se amb una antiguitat de 30 anys 11 mesos i un día, finalment ampliada a 31 anys "gràcies" a haver fet anys enrera 15 mesos de Servei Militar Obligatori (ves, qui ho havia de dir!), cada cop tenia més clar que el treball no quedaria enllestit en aquesta data. En vista d'això, va trobar prudent anar preparant el terreny: en primer lloc es va voler assegurar que, per part d'*UPCommons*, no hi haurien restriccions derivades del fet que l'autor d'una nova publicació ja hagués passat a millor vida (no en sentit metafòric sinó real), i en segon lloc calia informar el Departament de les circumstàncies que segurament l'obligarien a romandre en el despatx compartit uns quants mesos més enllà de l'u de setembre de 2009, amb el benentès que la seva permanència estaria condicionada a la disponibilitat plena d'espai, mobiliari i equip informàtic suficient per als companys que restaven al peu del canó. Amb benevolència per part d'uns, reticència per part d'altres i indiferència per part de la majoria, el qui a primers de març de 2010 escriu aquestes línies, segueix maldant per acabar el patracol, conscient que la benevolència copsada quan va posar Nadal com a data límit, ja derivava cap a la condescendència quan va demanar dos mesos més, i que la impaciència d'ara pot precipitar qualsevol dia en una amistosa invitació a fotre el camp d'una vegada.

Segurament no s'acabaria d'entendre tanta contumàcia sense fer esment d'un factor decisiu: la necessitat de seguir amb el mateix ordinador (el més senzillet dels 5 que hi ha en un despatx amb tres persones: qui escriu això i dues més), perquè el treball s'ha realitzat sobre la versió 16 d'AutoCAD (2004, comercialment), amb una llicència individual (no gestionada per cap servidor) com a professor de l'extinta assignatura ELEMENTS DE CAD, que fins el passat curs 2008/09 s'impartia a l'E.T.S. d'Enginyeria de Telecomunicació. I després del fotimer d'hores invertides, l'autor (que pot semblar agosarat en les seves manifestacions, però que és el súmmum del conservadurisme quan es tracta d'instal·lar, desinstal·lar o actualitzar *software*) no volia arriscar-se a traslladar-s'ho tot a un altre ordinador i quedar penjat o haver de seguir en unes condicions encara més precàries. Si no hagués estat això, us pot ben assegurar que a hores d'ara probablement faria el mateix que fa, però en un ambient força més distès, sense la vaga sensació de ser tolerat (freudians dogmàtics, absteniu-vos-en!: l'autor potser amaga mala consciència en algun racó de la seva vida, però no pas en aquest) i sense haver de posar cara de pòquer cada cop que li recorden que el Departament li fa un favor però que no és bo abusar-ne. N'ha hagut d'escoltar de tots colors. Fins i tot, en boca de buròcrates obedients, que això de seguir assistint al lloc de treball quan ja no toca no pot ser de cap de les maneres, i que tard o d'hora caldrà NORMALITZAR el tema per evitar aquestes transgressions. En sentir coses així, l'autor pensava: ¿quina Universitat Pública és aquesta en què, lluny de justificar la productivitat verificant l'assistència del personal en actiu (que probablement no és condició suficient però sí condició necessària, si més no davant del contribuent), consideren més important assegurar-se la no assistència dels jubilats? I li venien a la memòria aquelles ocasions en què, l'endemà d'alguna jornada de vaga amb prou ressò mediàtic, el Vicerectorat de Personal Acadèmic instava tothom a pronunciar-se sobre la seva participació, tot reconeixent implícitament la més absoluta ignorància del comportament laboral del personal en nòmina. El més curiós és que no preguntaven si el dia tal havies anat a treballar, sinó si havies seguit la vaga, probablement per entendre que una cosa era l'absentisme consuetudinari i una altra de diferent l'absentisme bel·ligerant.

És cert que aquesta obsessiva reticència a veure professors jubilats deambulant pels centres universitaris té o tenia el seu fonament: l'autor encara sap d'algun professor que després de jubilar-se com a funcionari seguia fent feinetes, però no sent prou continuades com per justificar l'obertura d'un despatx professional ni trobant a casa seva condicions per treballar i/o rebre clients, utilitzava més o menys esporàdicament les dependències departamentals, però és clar que això no té res a veure amb qui es va donar de baixa fiscal com a professional liberal al cap d'un any de sumar-se a la plantilla docent de l'E.T.S.E.T.B. (1983) i va causar baixa al seu col·legi professional tan aviat com va passar a Titular d'Universitat a temps complet (1989). Aquestes eren les petites corruptel·les pròpies d'alguns centres, com les escoles d'Enginyeria. A d'altres, com les d'Arquitectura, eren i són corrents pràctiques un pèl diferents: com que els despatxos d'arquitectura no acostumen a ser unipersonals i mai no passaria desapercebuda la presència habitual d'efectius aliens a la Universitat, encara hi ha professors amb dedicació a temps complet (pocs professionals motivats per la docència van optar per passar a temps parcial) que distribueixen la jornada entre l'aula universitària i el seu despatx; paradoxalment, aquest modus vivendi només mereix la reprobació de les autoritats si es practica sense pagar-li la butlla al Centre de Transferencia de Tecnologia (respecte al púdicament anomenat *overhead*, a la pàgina 6 de *Manipulació geomètrica dels blocs d'AutoCAD*, d'aquesta mateixa col·lecció, l'autor formula una proposta), com si projectar un edifici de tres plantes entre mitgeres o redactar un informe sobre infiltracions a través d'un terrat esquerdat encaixés en una categoria tan altisonant com semànticament inadequada (TT, un eufemisme que l'autor sempre s'ha resistit a escriure amb totes les lletres, tret de quan s'ha de referir al Centre abans esmentat). Si realment hi hagués voluntat de superar un posicionament que, més enllà d'ambigu podríem titllar de farisaic, sobren vies per estar al corrent de l'activitat extraacadèmica del personal docent: requisitòries als col·legis de llicenciats, enginyers i arquitectes, per tal que facilitin estat d'afiliació i dades sobre feines visades o, si això no oferís prou garanties de veracitat perquè la complicitat gremial preval sobre el civisme, l'exigència a l'interessat d'una autorització perquè la Universitat pugui accedir a la informació que sobre ell té l'Agència Tributària (si l'Entitat Metropolitana del Transport -de Barcelona- la demana per a la concessió de la Targeta Rosa, ¿per què la Universitat no hauria de fer-ho, vetllant per l'estricte compliment de compromisos lliurement adquirits?).

Tot això, sens dubte, escandalitzarà molts d'aquells professors funcionaris que no s'escandalitzarien gens si se'ls demostrés, dividint la seva retribució entre les hores dedicades de debò a la Universitat, que estan més ben pagats que un ministre i que viuen força millor. Com a funcionari que es limitava complir amb les seves obligacions, tractant de compensar amb un plus de dedicació les seves limitacions com a comunicador (parlant sense embuts: la seva condició de professor mediocre), l'autor podria contemplar amb indulgència el mal exemple de la no tan minoritària minoria objecte d'aquesta diatriba, i sobretot mossegar-se la llèngua, per allò de *la ropa sucia se lava en casa*. Però com a contribuent, quan pensa que el pagament d'hores improductives o cobrades per partida doble també surt de la seva butxaca, renega de l'*omertà* i es creu en el deure d'esbombar-ho tot. A tall d'anecdota, per cloure d'una vegada aquesta primera ràfega indiscriminada (cloenda anunciada ja fa dues pàgines, en finalitzar el paràgraf inicial, però encara no consumada) l'autor referirà com l'ingrés en el Cos de Funcionaris va suposar una revelació penosa que el va fer madurar de cop en un aspecte en què havia quedat encarcarat, com qui diu anant-se a dormir trotskista i llevant-se a l'endemà convertit en socialdemòcrata. Des de petitet ja li havien ensenyat que els règims polítics dels països anomenats socialistes tenien poc a veure amb el socialisme, però en el fons encara creia que aquelles societats havien donat un primer pas (expropiar els mitjans de producció a la burgesia) i que només quedava arrabassar el poder a la burocràcia mitjançant una revolució política, esdeveniment que sens dubte es produiria en el marc de la revolució mundial, en què als països capitalistes els esperava una doble tasca: la revolució política i la social. No li fa cap vergonya reconèixer la seva adhesió a aquest ideari, en tot cas menys pueril que l'arnada mitologia lliuremercadista a què desesperadament es van aferrar les masses del primer grup de països, quan tot se'ls va anar en orris. No va ser únicament el fracàs de la *perestroika* a la URSS, propiciat tant pel boicot dels *aparatxic* com per Occident, tot en el context d'una economia virtualment devastada, i la consegüent ensulsiada dels països satèl·lits, allò que li va fer obrir els ulls, sinó l'extrema rapidesa amb què els aspirants a una plaça abandonaven tota vel·leitat stakhanovista tan bon punt aconseguien la poltrona, experiència que li va aportar l'argument definitiu per adonar-se de la inviabilitat de sistemes que pretenen garantir treball, salut i cultura a tothom, estatalitzant l'economia: UNA SOCIETAT DE FUNCIONARIS NO FUNCIONARÀ MAI A LA VIDA!

En les tres pàgines que porta, el lector ja se n'haurà adonat que aquesta és una *Introducció* vindicativa, que l'autor ha aprofitat per treure's diverses espines. Però, en l'autoencomanada missió de desmuntar tòpics amb altres tòpics, tampoc no voldria fer-se pesat, així que es limitarà a referir dues anècdotes més, amb les quals creu que n'hi haurà prou per desfer el mite d'una Universitat al marge de les misèries del segle, on tampoc no és or tot el que lluu, i es compromet a fer-ho sense recórrer al mot "excel·lència", sobreutilitzat aquests darrers anys, ni al de moda enguany, no només superflu, tenint "govern" i "governació", sinó del tot incorrecte però d'imparable implantació entre la classe política i acadèmica (és a dir, funcionarial): "governança".

La primera té a veure amb la Tesi Doctoral *Elementos para una exploración dinàmica de la forma visual*, que l'autor va perpetrar entre els anys 1982 i 1986. El títol enfàtic no té massa a veure amb el contingut (creació d'un programa de modelatge i visualització d'objectes i escenes 3D, en un moment en què a Espanya encara no es comercialitzaven productes així), però com que l'autor era arquitecte (entre 1971 i 1982 va intentar exercir-ne, sensa massa fortuna) i, segons les lleis vigents a l'època li tocava presentar-la a l'E.T.S. d'Arquitectura de Barcelona, malgrat que havia deixat l'ofici i treballava de professor de dibuix a l'E.T.S. d'Enginyeria de Telecomunicació, va caldre "vestir" el tema si més no pel que fa al títol. Un altre concessió a la galeria va ser exhibir una nodrida bibliografia que mai no va utilitzar ni necessitar per dur a bon port el seu treball. Què hi farem! Ningú no és perfecte i aquesta era i és una de les seves debilitats: donar preferència a la satisfacció personal de saber-se capaç de redescobrir la sopa d'all amb els seus propis mitjans (tenacitat, paciència i la disposició a deixar-hi moltes hores) per damunt de la possibilitat de donar un petit pas endavant respecte a realitzacions d'altri divulgades mitjançant publicacions, camí que sens dubte és el del progrés.

Però, reprenent el fil de la narració, per situar-la en el seu marc històric direm que tres companys de carrera i gairebé de promoció van decidir adquirir a mitges un PC (de fet, en 1982 encara no s'havia encunyat el terme "PC", ni tan sols el de "compatible IBM" que el va precedir) que avui faria riure, amb la lloable intenció de desenvolupar les seves respectives tesis doctorals compartint l'equip i també les troballes i experiències. Abans de seguir, convindrà atribuir noms a aquests tres doctorands, i direm que el senyor M era l'únic que compaginava l'exercici de la professió d'arquitecte amb la docència a la U.P.C., concretament a l'E.T.S.A.B. i en el departament que més endavant seria Expressió Gràfica Arquitectònica I, raó per la qual inicialment volia orientar la seva Tesi cap a la incidència pedagògica del CAD en l'ensenyament del dibuix arquitectònic. Els altres dos, els senyors J i K (darrera d'aquesta K s'amaga qui us dóna la tabarra, que sempre s'ha sentit més proper a Karl Marx que a Carlo Borromeo), havien plegat d'arquitectes i exercien de professors de Dibuix Tècnic a l'E.T.S.E.T.B., en el que acabaria sent la secció més petita del Departament d'Expressió Gràfica a l'Enginyeria. Menys dotats que el Sr. M per a les matemàtiques i la informàtica, la situació familiar i la més gran disponibilitat de temps dels senyors J i K els permetia assumir, en canvi, el risc d'embrancar-se en una aventura que els atreia, i van optar per una orientació més "tècnica" en contraposició al caire més "humanístic" del tema escollit pel primer: cadascú en consonància amb les pròpies habilitats i mancances, i seguint línies metodològiques independents, entre els seus programes hi havia inevitables punts de contacte però diferien prou en les fites que s'havien proposat com per no ser presos com a simples variacions sobre un mateix patró; tanmateix, quedava clar que entre els dos àmbits d'actuació (gairebé fronterers i amb nombroses zones que eren "terra de ningú") ja no quedava espai per una tercera incursió, que sense voler-ho s'hauria encavalcat amb alguna de les les dues propostes o amb les dues alhora. Doncs això va ser precisament el que va succeir, quan el Sr. M va adonar-se que el tema previst no només era més espinós que no semblava d'entrada, si es volia fer amb un mínim de rigor, sinó que hi havia massa risc d'entrar en confrontació amb sectors de l'E.T.S.A.B. que podien veure en perill els seus interessos. A més, el debat cultural no era el camp en què se sentia més segur, almenys no tan segur com en el domini de la programació informàtica, així que va decidir recuperar el temps perdut (en aquell moment, els senyors J i K duien més de dos anys treballant en les respectives tesis) i plantar la tenda entre les parcel·les J i K, conscient de comptar amb un gran avantatge: a diferència dels Srs. J i K, ell jugava a casa i, amb poca cosa que fes (en aquell Centre encara no s'havia llegit cap Tesi sobre Informàtica Gràfica), el seu treball seria ben rebut i podria aspirar a una bona qualificació; a condició, es clar, de presentar-lo abans que els senyors J i K. Ja se sap: tots som iguals, però uns són més iguals que altres (George Orwell *dixit*).

Potser pel seu tarannà, aquests darrers mai no s'hauríen aprofitat d'una situació d'avantatge amb la desimboltura amb què ho va fer el Sr. M, però en aquella època (ara ja no m'atreviria a assegurar-ho) el Sr. J era més càndid i el Sr. K (qui us parla), sense ser un murri, no era tan fàcil d'entabanar. Així que la seva reacció va ser diferent: mentre el Sr. J va seguir amb el programa de treball que s'havia marcat, impertorbable, el Sr. K va pensar que l'única manera d'evitar que el dia de la seva Lectura de Tesi algú del Tribunal pogués adduir que mesos abans ja se n'havia llegit una de molt semblant, en aquella santa casa (l'E.T.S.A.T.B), era afanyar-se i presentar el seu treball (estigués com estigués) el mateix dia que ho fes el Sr. M. Va demanar al seu Director de Tesi que preparés el preceptiu informe i a corre-cuita va enllestir dos patracols, dels quals va dipositar dos exemplars: un a la Biblioteca de la E.T.S.A.B. i l'altre en el Rectorat. Durant els mesos que van transcórrer fins a la Lectura (juliol de 1986), va tenir ocasió de repassar-s'ho tot amb deteniment i va trobar nombroses pífies, fruits de la precipitació. Com que l'operació de substituir les pàgines afectades hagués estat problemàtica, va optar per afegir al segon volum un annex de 31 pagines amb 5 esmenes (*ENMIENDAS Nº 1 a Nº 5*, a més d'una *NOTA PARA LOS USUARIOS DE APPLE-II*), que restablia el bon ordre del contingut. Els exemplars que va lliurar al Director, Dr. Manuel Medina, i als cinc membres del Tribunal, doctors Manuel Baquero, Pere Brunet, José Antonio Franco, José María Gentil i Joan Trias, ja incorporaven aquest annex. Tampoc hi va haver dificultat a afegir-les a l'exemplar de la Biblioteca de la E.T.S.A.B. però, llegida la Tesi, li va fer mandra (o vergonya) anar al Rectorat per fer el mateix.

Els fets van mostrar que el realisme del Sr. K tenia més fonament que l'idealisme del Sr. J, perquè quan aquest va llegir la seva Tesi (més meritòria que la del Sr. K i molt més que la del Sr. M), quatre mesos més tard, se'n va sortir amb l'Apte *Cum Laude* (llavors es donava per suposat aquest reconeixement, com el valor a la Mili; si no te l'atorgaven, era com una escopinada a la cara), però pels pèls: la deliberació del Tribunal va ser anormalment llarga i les inevitables filtracions deixaven entreveure reticències per part d'un distingit membre, el més expert en la matèria, que també ho havia estat en la Lectura de Tesi del Sr. K (però no en la del Sr. M, sortosament per a ell). Però bé, *all's well that ends well*, i tampoc no és aquesta la raó per la qual l'autor (el Sr. K de la història) ha portat fins aquí el relat, sinó precisament pel que ve ara. Vint anys després, conscient que la seva Tesi no passaria a la posteritat (en el registre de la Biblioteca de l'E. T.S.A.B., només constava un despistat que l'hagués consultat, segurament buscant una altra cosa) però incòmode davant la improbable cadena de supòsits 1) que algú demanés l'exemplar del Rectorat; 2) que se'l llegís, i 3) que localitzés les cinc errades, va voler aprofitar bones coneixences al més alt nivell per veure què s'hi podia fer amb aquest exemplar, que mentrestant havia passat a una secció especial de la nova Biblioteca Rector Gabriel Ferraté del Campus Nord. Aquest contacte va contactar al seu torn amb el Director de l'Àrea de Recerca, que era qui havia de saber més d'aquestes coses, el qual li va suggerir que contactés una altra persona de la Unitat Tècnica de Gestió de 3r Cicle, dient-li que des d'allà recuperarien els exemplars de les dues biblioteques (la de l'E. T.S.A.B. i la de Rector Gabriel Ferraté), comprovarien si les esmenes del primer eren les que havia valorat el Tribunal de la Tesi (¿com, si no era preguntant-ho a aquests cinc doctors, un d'ells jubilat i dos residents fora de Catalunya?) i les incorporarien al segon. L'autor ha pogut precisar aquestes gestions perquè conservava un e-mail, però de la resta ja només li queda el difús record d'un seguit de trucades telefòniques i d'anades i vingudes, fins que va dir: prou! No era possible de franquejar una tan ben travada barrera burocràtica i finalment va desistir, perquè si seguia bregant encara s'exposava a que tot acabés amb l'eliminació de les esmenes incorporades a l'exemplar d'Arquitectura. No n'està del tot segur, però creu recordar que algú li va suggerir que una possible via de solució era digitalitzar la Tesi (llavors era l'interessat qui ho havia de fer), però tampoc es descartava que un cop fet sorgís el mateix conflicte (decidir si la versió "autèntica" era la primera o l'esmenada) i va pensar que si havia aconseguit sobreviure vint anys amb la lleu insatisfacció de saber que en un domini públic hi figurava la seva Tesi amb 5 errors, segurament en podria seguir vivint uns quants més.

Aquesta és una mostra de com burocratisme i defensa aferrissada del progrés són termes antagònics (i ara que la burocràcia de l'Espai Bolonya se suma a l'estatal, autonòmica i local, que el Déu catòlic ens agafi confessats!). Serà una minúcia irrellevant, que l'autor (que pel cap baix arrossega la creu aquests cinc errors) no voldria pas que fos interpretada com un atac personal a un dels cervells més ben moblats del socialisme espanyol (i conspicu representant de la majoritària ala

jacobina del partit), però l'actual Rector de l'Institut Universitari Europeu de Florència, Dr. Josep Borrell, de qui aquest admira més els seus articles d'opinió com economista en EL PERIÓDICO DE CATALUNYA que la crònica dominical en el mateix diari *Carta des de Florència*, el va deixar perplex quan la del diumenge 8/11/2009, titulada *Un palau per al saber* (pàgina 17) començava amb aquesta frase, a propòsit de l'estada d'Stendhal a la ciutat en 1817 i l'aclaparament que el va envair a la vista de tanta saturació artística (reacció homologada anys després com a *síndrome d'Stendhal*): "Contemplant Florència des dels turons de Fiesole, a la tèbia llum dels últims sols de tardor, és fàcil sentir la síndrome de Stendhal, aquella malaltia de l'ànima que **el romàntic alemany** va patir al descobrir la bellesa d'Itàlia". Que el lector tregui les seves pròpies conclusions, de la perla que el Sr. K ha destacat en negreta, i passarem full, perquè la segona anècdota que volia referir no era pas aquesta sinò un episodi de la història no escrita de la U.P.C., no tan representatiu dels mals del burocratisme com de l'existència en aquesta Universitat (i probablement a totes) de lluites mig soterrades però ben sagnants (entre centres, entre departaments, entre seccions departamentals, entre càtedres, entre capelletes i entre individus; pels diners i altres recursos, pel manteniment de situacions de privilegi o per abastar àrees cada cop més àmplies de poder): en definitiva, "com a la vida real" o, si voleu, lluites "reals com la vida mateixa", utilitzant un llenguatge poc científic però perfectament entenedor. Però, ¿que tot això passa no és obvi? Doncs potser no tant, perquè l'opinió del carrer de vegades és tremendament ingènua i sovint la Universitat, expressió màxima de la Cultura i Seu de l'Art, la Ciència i la Tècnica, on a tothom se li dóna veu, les discussions sempre es desenvolupen de forma civilitzade i només incidentalment algú es deixa arrossegar per la passió del debat científic, és objecte d'una mitificació de la qual avui no gaudeixen ni el poders legitimats (Legislatiu, Executiu i Judicial) ni els fàctics (ni l'Econòmic, ni l'anomenat Quart Poder, ni tan sols l'Església). De la narració que segueix podreu deduir com sodomitzant les persones compromeses amb la funció principal de la Universitat, que és transmetre coneixements (crear-ne de nous en serà la segona, però l'autor no creu que mercadejar amb aquests nous coneixements n'hagi de ser la tercera, almenys de la Universitat Pública), ni que sigui disfressant-ho amb invocacions a millores en l'eficiència, gairebé sempre en surt perjudicada la qualitat d'aquesta funció.

En aquest episodi el repartiment s'amplia a J, K, L i M. En realitat, el Sr. M fa mutis i ja no intervé més, els Srs. J i K segueixen, i s'hi incorpora la Sra. L. També hi surten altres personatges que, pel menor protagonisme en els fets que es relaten, no hi haurà problema a citar pel seu nom. Primera d'aquesta especialitat a Catalunya, l'E.T.S. d'Enginyers de Telecomunicació de Barcelona va començar a rutllar a Terrassa (això de Barcelona, doncs, es devia més al fet que la U.P.C. es digués encara U.P.B. que a la seva ubicació) en el curs 1971/72 i al cap de quatre cursos va saltar a Barcelona, on va ocupar diversos locals (tots a l'entorn de la *Jefatura Superior de Policía*), fins que en el curs 1978/79 va poder estrenar una construcció prefabricada que, malgrat la seva provisionalitat, reunia els mínims estàndards i va seguir funcionant fins el curs 1993/94. Ja va ser aquesta la llar (situada entre els actuals edificis E-6 i Nexus II del Campus Nord i el bosquet on ara hi ha el Rectorat) que va acollir els nous professors J (curs 1978/79), K i L (curs 1981/82). Els pioners Josep M. Massot, Joan M. Pavía, Miquel Donada, Miguel A. Laguéns i Ignasi Miquel ja no hi eren quan van arribar els dos últims (tret del primer, que durant uns anys va seguir exercint una certa tutela a distància) però sí que per deu cursos més van poder seguir gaudint de l'experiència i la companyia d'un veterà: Xavier Baladia, personatge polifacètic (ell s'estimava més parlar de "un tot terreny") que, en opinió de l'autor, era l'ànima d'un Grup de Càtedra IV (encara estava per fundar-se el Departament d'Expressió Gràfica a l'Enginyeria) considerat a l'escola amb un respecte no exempt de certa reticència, perquè des de sempre s'havia catalogat l'assignatura DIBUIX TÈCNIC (1r Curs, Pla d'Estudis 1964) com una relíquia del passat (per a què necessita un enginyer de telecomunicació el dibuix? era el que pocs s'atrevien a preguntar però que llegies en tots els ulls).

Per a una persona com el Sr. K, que encara tenia obertes les nafres del seu pas per l'E.T.S.A.B. on, amb un petit encàrrec de curs, havia tingut ocasió de copsar l'ambient poc fraternal d'aquella santa casa i en concret d'un departament que li havia fet venir a la memòria aquella secció del setmanari humorístic *LA CODORNIZ* anomenada *La oficina siniestra* (inclosos els poc entranyables personatges *Pelota nº 1*, *Pelota nº 2*, etc.), l'esperit d'obertura i cooperació interdisciplinar de l'E.T.S.E.T.B. va suposar una alenada d'aire fresc, malgrat el fràgil encaix de l'equip de professors en l'estructura del centre. Potser justament perquè eren tan insignificants que no representaven cap perill per als col·lectius que hi tallaven

el bacallà, se'ls deixava més llibertat del que era habitual. I, potser perquè l'oportunitat d'integrar-se en aquell equip coincidia amb un moment en què estava tocant fons, professionalment i també personal, el Sr. K va capgirar en poc temps l'adversa reacció inicial (mai més a la Universitat!) per una disponibilitat total i entusiasta: als seus companys se'ls veia bona gent i en aquella escola hi havia possibilitat de fer coses noves. Fins i tot l'exigència peremptòria de fer la Tesi Doctoral, en lloc de llastar l'activitat docent (per recórrer abusivament al pilot automàtic de la rutina) l'estimulava, amb la incorporació de nous punts de vista. Curiosament, al Sr. J li passava tres quarts del mateix. I és que l'E.T.S.E.T.B., en aquella etapa de creixement prenyada de bons auguris, exercia poder d'atracció fins i tot sobre dos individualistes com els Srs. J i K, llops esteparis cadascun en el seu estil. La Sra. L, menys neuròtica i més com cal, queda una mica al marge del que explicarem, per una major capacitat d'adaptació que li va permetre sentir-se menys afectada pels canvis i resituar-se gradualment en el Departament passant a l'E.T.S.E.I.B. (destí que tampoc no podia eludir, a diferència dels Srs. J i K, l'edat més avançada dels quals els ha permès optar per la jubilació voluntària).

Aquesta arcàdica situació, en què imperceptiblement les ingenuitats fundacionals (entre tots ho farem tot) anaven quedant enrera per donar pas al burocratisme que probablement sigui peatge inevitable de tot procés de consolidació institucional, es va acabar de tòrcer per influència de tres factors: la prevista estructuració de la U.P.C. en departaments; la construcció del Campus Nord que, d'acord amb el punt precedent, s'havia projectat com a conjunt de petits edificis dedicats a la docència (aularis), als departaments i als equips administratius de tres centres (Facultat d'Informàtica, i EE.TT.SS. d'Enginyers de Telecomunicació i d'Enginyers de Camins, Canals i Ports); la renovació d'uns plans d'estudis que, pel que fa a l'àmbit de les Telecomunicacions, va fracassar en l'intent de dur fins les últimes conseqüències l'organització dels estudis/títols en una seqüència única de cicles, i va seguir arrossegant la separació curricular i d'ubicació física entre centres de grau mitjà (les Enginyeries Tècniques es van situar provisionalment a Sant Just Desvern, esperant assentar-se definitivament en el futur Campus de Castelldefels) i de grau superior (Campus Nord). Però, ja abans que aquests tres canvis es fessin realitat, va esdevenir-se un fet premonitori: la marxa del professor Baladia, que en opinió dels Srs. J i K va constituir un error de càlcul que no el va beneficiar a ell ni a la resta del grup. Tot va començar amb la dècada dels noranta, quan els Srs. J i K, sens dubte influïts per les respectives tesis i amb la confiança que dóna saber que, investits funcionaris, eren gairebé invulnerables, van decidir que els estudiants ja no podien esperar al Nou Pla d'Estudis (que havia de reconèixer la necessitat de més hores i més mitjans per a l'assignatura gràfica: fins aquest punt tenien una bena davant dels ulls!) i calia formar-los en la utilització dels sistemes CAD. El Departament d'Expressió Gràfica a l'Enginyeria, tot just fundat, ho veia massa agosarat, però a la Comissió d'Estudis de l'E.T.S.E.T.B. li semblava perfecte sempre que els nous continguts docents fossin compatibles amb la dotació del centre (15 PCs amb sortida gràfica per a més de 600 alumnes!). Li estalviarem al lector l'ensurt de saber en quines condicions s'impartien les classes teòriques de CAD, com es feien les pràctiques i com s'avaluava tot plegat, i tornarem a la qüestió de la precipitada decisió del professor Baladia: potser perquè l'exercici de la professió d'arquitecte (a diferència dels professors J, K i L, mantenia un despatx) li deixava menys temps, se li feia una muntanya aprendre l'AutoCAD, cosa que poc temps després no va tenir més remei que fer, com tots els arquitectes que havien d'acabar delineant-se els plànols, i va creure que el millor era anar-se'n. Tot sumant-se a la desafecció (eufemisme força emprat enguany a Catalunya) que els fets que ara explicarem van provocar en la resta de l'equip, singularment en els Srs. J i K (ambdós propensos al solipsisme), l'absència d'aquest cohesionador nat va propiciar el desarrelament, desfermant les tendències centrífugues subjacents.

Tot estava prou justificat, si es vol, però va molestar la barroeria amb què en el Document 19/8, aprovat a la Junta de Govern del 3 de novembre de 1989, s'intentava daurar la píndola a quatre seccions departamentals de l'E.T.S.E.T.B. (Enginyeria Química, Expressió Gràfica a l'Enginyeria, Organització d'Empreses i Anglès -així era com anomenàvem tots el 4t grup, integrat a l'heterogeni departament Projectes d'Enginyeria), per tal que acceptessin de bon grat el desterrament a l'edifici de l'E.T.S. d'Enginyers Industrials de Barcelona (Campus Sud), compartint espai físic amb les seccions departamentals locals mentre l'encàrrec docent seguia vinculat al Campus Nord: "consolidar la vida departamental" o bé "aconseguir la massa crítica necessària per a desenvolupar el treball docent i de recerca" van ser alguns dels pietosos arguments que s'hi esgrimien. L'execució del trasllat forcós a l'E.T.S.E.I.B. va tenir lloc en juliol de 1994, però mentrestant no van parar de succeir-se

negociacions a múltiples bandes, que a uns els van servir per fer-se un lloc a les noves instal·lacions del Campus Nord (cas dels companys d'Organització d'Empreses, que a diferència de la resta consolidaven i augmentaven la seva presència en els nous plans d'estudis, i de les dues professores d'Anglès, de fàcil reubicació), i als demés (Enginyeria Química i Expressió Gràfica a l'Enginyeria) sols per ajornar l'aplicació de la sentència fins a pràcticament la vigília de l'enderroc d'aquella construcció provisional que durant 15 anys havia estat la seu de l'E.T.E.T.B. Cal aclarir que el Director del centre (que en la data d'execució havia passat a ser Vicerector d'Ordenació Acadèmica) almenys va tenir la gallardia de donar la cara, defensant la procedència del trasllat, i va contribuir personalment a millorar el confort del nou despatx; els antics companys, que gradualment havien anat passant a ocupar les noves dependències del Campus Nord, guanyant espai (quantitativament i qualitativa) i no mostrant cap mena d'interès en renunciar ni a un metre quadrat d'aquest guany, van mantenir un ambigu equilibri entre el recolzament moral i cert fatalisme vergonyant; el departament ni es va pronunciar sobre el tema: potser era que s'ho creien, allò de "consolidar la vida departamental". En realitat, no tan sols no va arribar a assolir-se cap "massa crítica" per una major interacció entre els professors d'Industrials i els de Telecomunicacions (a excepció de la Sra. L que, com ja s'ha dit, va acabar integrant-se en el primer grup en produir-s'hi les primeres jubilacions) sinó que el segon grup va desmembrar-se virtualment (parlem de 3 persones!), desenllaç clarament predictible després de la marxa del professor Baladia, i que l'atomització de l'oferta docent i l'allunyament entre el despatx i l'aula (de tot això en parlarem de seguida) encara van accelerar. No cal dir que el rendiment acadèmic dels Srs. J i K també se'n va ressentir: si no la qualitat de la docència (que si fa no fa va seguir igual: més alta la del Sr. J que la del Sr. K), perquè tots dos tenien punt d'honor, sí pel que fa a l'interès a renovar els continguts (sort que al Sr. J l'engrescava tant l'evolució dels programes que ensenyava com posar-ne al corrent els alumnes, perquè, no podent programar, al Sr. K més aviat li resultava una obligació tediosa incorporar a la docència els canvis aportats periòdicament per les successives versions) i en la participació en les instàncies de discussió i/o decisió a nivell d'Escola, Departament i Universitat (el Sr. K, que sempre havia estat el més compromès, a partir de l'any 1994 se n'hi va anar desvinculant perquè sentia que res de tot allò ja no era cosa seva).

Si hem començat el paràgraf precedent confessant que potser el trasllat no era del tot injustificat, objectivament parlant, és perquè les forces vives de la U.P.C. ja estaven assabentades de la troncalitat dels nous plans d'estudis (i també, en conseqüència, d'aquelles matèries que en serien bandejades, tret d'escorrialles en forma d'assignatures optatives o de lliure elecció) quan a finals de 1989 es van posar a partir el pastís del Campus Nord. I és que els Plans d'Estudis de l'àmbit de les Telecomunicacions (que oficialment van veure la llum en 1992), com és lògic i al marge del que volguessin alguns somniatruites, recollien velles aspiracions com la d'eliminar el Dibuix. Una altra història que tothom pot entendre és que, al marge del coneixement més extens i rigorós que n'hagin de tenir Arquitectes, Enginyers de Camins o Industrials especialitzats en segons quines branques, igual com qualsevol ciutadà té el dret i el deure de ser alfabetitzat, per bé que acabi fent de manobre, hauria de tenir una capacitació mínima per interpretar i generar documentació gràfica senzilla: per exemple, la que hagi de permetre a una persona realitzar el croquis d'un armari adaptat a un racó oblic de casa seva, acotat de forma inequívoca, o bé comprovar si l'armari representat en un croquis que se li facilita s'adapta a aquell racó. Tothom acceptarà que aquesta mínima capacitació gràfica s'hauria de garantir en l'ensenyament obligatori, i és aquí on sorgeix el conflicte, perquè és notori que molts estudiants accedeixen a la Universitat sense aquest requisit (com ho és el grau d'alfabetització d'alguns, més que discutible), però si és cert que la Universitat no es pot permetre el luxe de fer tasques de suplència quan els mitjans de què disposa estan al límit d'allò que se li exigeix, també ho és que no hauria d'acceptar candidats amb mancances tan greus. Tot això, per acabar concloent dues coses: que sort en van tenir els professors de DIBUIX TÈCNIC de la iniciativa d'haver incorporat nocions de CAD a l'assignatura en vies d'extinció, demostrant així que no només estaven capacitats per impartir Geometria Descriptiva i obrint la porta a que se'ls concedissin unes engrunes en forma d'una Optativa de Primer Cicle, ELEMENTS DE CAD, i l'Assignatura de Lliure Elecció (ALE) ANIMACIÓN EN 3D que 15 anys després (quan, al seu torn, els Plans d'Estudis del 92 entraven en la recta final, obeint als dictats de Bolonya) es desdoblaria en dues per adaptar-se a una demanda minvant; que, malgrat això, el fet que que tothom que va passar a un Campus Nord acabat d'estrenar hagués doblat l'espai, pel cap baix, no va obstar perquè les sentències de confinament en el Campus Sud s'acomplíssin de forma inexorable.

Ja que parlàvem de les engrunes, acabarem de precisar els continguts d'aquestes assignatures, ni que sigui per deixar-ne alguna constància quan el vent s'endugui tot altre rastre. L'E.T.S.E.T.B. habia concedit el *nihil obstat* a la primera amb la condició que almenys una quarta part del temps es dediqués al CAD electrònic (OrCAD), raó per la qual va caldre comprimir el CAD de propòsit general (AutoCAD) de 60 a 45 hores. Es va acceptar, tot i la convicció que no tenia massa sentit una aproximació purament gràfica al disseny de circuïts, al marge de l'estudi del seu comportament físic i deixant de banda els mòduls de simulació, però per fer això hagués calgut una col·laboració interdepartamental que mai no es va produir perquè cap de les dues parts interessades la volia realment: de mica en mica es va anar abandonant aquesta part (abans a l'escola de Barcelona que a la de Sant Just, on també es donava l'assignatura amb petites variacions quant proporció de contingut) a favor de l'AutoCAD, de primer perquè també s'estimava profitós per a l'estudiant que conegués mínimament les tècniques de renderització de la imatge, i després per l'imparable inflació de les possibilitats del sistema, des de la versió 10 amb què es va començar fins a la 16 (ACAD 2004), emprada quan es va acabar (curs 2008/09). Aquí el Sr. K (és a dir, l'autor), responsable d'aquesta assignatura, aprofitarà l'avinentesa per revelar una petita frustració: com que la Facultat d'Informàtica portava un any d'avantatge a l'E.T.S.E.T.B. en la implementació dels Nous Plans d'Estudis però encara no disposava d'una oferta massa àmplia d'ALEs, en el curs 1993/94, primer i segon quadrimestre, se n'hi va impartir una que era l'optativa 1r cicle prevista per a l'E.T.S.E.T.B. però substituint l'Orcad per una iniciació a la programació en AutoLISP; el Sr. K xalava com mai ho havia fet donant classe, perquè per primer cop podia jugar al joc que més l'entretenia; però el miratge va ser efímer perquè, quan a partir d'aquesta experiència va suggerir que una de les ALEs que podien oferir (oferir i no "ofertar") des de l'E.T.S.E.T.B. fos justament PROGRAMACIÓ EN AUTOLISP (restringida a qui ja conegués l'AutoCAD, per haver cursat prèviament ELEMENTS DE CAD o per haver-ne après de qualsevol altra manera), els seus companys J i L li van treure del cap argumentant, probablement carregats de raó, que una ALE no havia de comportar massa especialització i que havia de tenir una càrrega lúdica (per als alumnes, no per al professor) que aquella proposta no tenia. Desil·lusionat, i vist que el Sr. J es divertia més devorant llibres sobre 3D-Studio (aquest era el programa en què es basava ANIMACIÓN EN 3D) i no li venia massa de gust col·laborar amb el Sr. K en la creació de curtmetratges d'animació (a diferència del Sr. J, de tarannà estudiós i que si hagués pogut viure de rendes s'hauria passat la vida cursant carreres universitàries, una darrera l'altra, el Sr. K ja frisava per posar en pràctica les quatre primeres coses mal apreses), va començar a desenvolupar programets pensats com exercicis per a aquella assignatura nonada, activitat que mica en mica va anar desvinculant de la pròpiament docent. Pel que fa a ANIMACIÓN EN 3D, en les sis últimes edicions es va replegar sobre les tècniques de visualització (igual com passava amb AutoCAD, a cada nova versió les possibilitats de 3D-Studio es multiplicaven) i les de modelatge de sòlids se'n van segregar, convertint-se en la nova assignatura TÉCNICAS DE MODELADO POR ORDENADOR. La versàtil Sra. L, no sols col·laborava amb el Sr. J amb aquestes ALEs sinó que es va responsabilitzar del disseny, sota demanda de l'E.T.S. d'Enginyers de Camins Canals i Ports, i de la impartició d'una assignatura muntada com ELEMENTS DE CAD sobre l'aprenentatge d'AutoCAD però que tenia tenia característiques específiques.

Si aquesta atomització de matèries ja propiciava la disgregació de l'equip, va ser l'allunyament de l'aula (i la diferent repercusió que això va tenir sobre els seus membres) el factor que la va precipitar, sense que la generosa aportació de l'E.T.S.E.T.B., posant a disposició de tots els "professors transeünts" un despatx comú en els aularis del Campus Nord, dotat amb un ordenador d'ús també compartit (no us penséssiu pas que era una contribució al confort dels professors, sinó una mesura per evitar-los als estudiants haver d'anar al Campus Sud per realitzar consultes), aconseguís deturar-la. Com que totes les assignatures de què hem parlat es feien en aules informàtiques i tenien un caràcter eminentment pràctic, en el cas que ens ocupa gairebé totes les consultes es realitzaven a classe i les permanències en el despatx *ad hoc* sí que s'haurien pogut convertir en una forma de pal·liar la pèrdua miserable de temps Diagonal amunt, Diagonal avall, però a l'hora de la veritat les coses no acabaven de funcionar, així que el Sr. J (que tenia a casa un ordinador millor que el del despatx de l'E.T.S.E.I.B., i no cal dir que millor que el del Campus Nord) matava els temps morts, entre classe i classe, a la Biblioteca Rector Gabriel Ferraté, i tan aviat com acabava l'última classe se'n tornava a casa seva; la soferta Sra. L va provar uns quants anys d'endur-se feina a aquest despatx però finalment va dessistir i s'estimava més tornar a l'E.T.S.E.I.B. o marxar-se a casa si ja era tard; el més tossut, potser perquè no li agradava endur-se feina a casa

i creia que s'havien de respectar les hores de permanència encara que només fos per allò de la muller del cèsar (ja ho enteneu: que no n'hi ha prou a ser honest sinó que cal donar-ne testimoni), era el Sr. K, que us pot ben jurar que en va quedar fart i que va decidir demanar la jubilació voluntària sobretot perquè no li semblava decent seguir exercint de professor fastiguejat com estava d'aquesta situació. Per més que s'esforcés cada dia a pensar què s'hauria d'endur a l'endemà per poder fer alguna cosa de profit en el Campus Nord entre classe i classe, tres dies de cada quatre s'oblidava algun paper o disquet i havia d'acabar posant-se a llegir el diari: ja ha explicat a l'inici d'aquesta introducció que així se li va despertar l'afició als sudokus que l'ha acabat portant a escriure aquestes línies. Preneu-ne nota, inefables gestors i maldestres auspiciadors de "masses crítiques"!

L'autor no voldria que tanta magnanimitat establís un precedent, però vol premiar tots aquells lectors que hagin arribar aquí sense defallir, amb un *bonus*: un dels programets que es dedicava a fer, per si prosperava el projecte de donar una ALE de programació en LISP però en el fons per evadir-se de la imminència del trasllat al Campus Sud; els servirà com il·lustració de l'últim relat i de passada, si els ve de gust, podran jugar-hi.

No només per aquest trasllat físic, sinó per tot una sèrie de variades vicissituds que van experimentar els arxius informàtics antics, els anteriors a l'any 2001 no conserven datació fiable. Però, lligant records dispersos, l'autor creu que aquest programa va ser creat en 1994, poc abans de l'exili, probablement amb la versió 12 d'un AutoCAD en anglès i encara en un entorn MS-DOS. Com que deu fer 15 anys que va deixar de mirar-se'l, li ha costat unes hores recordar com anava i, sobretot, fer els petits ajustos, que descriurem més endavant, per aconseguir que rutllés en un altre ordinador que, malgrat estar gairebé obsolet, és molt més ràpid que el de llavors, en una versió castellana d'AutoCAD 2004 i naturalment sobre Windows (XP). Suposa (però, francament, ara no ho podria assegurar) que la intenció devia ser muntar programa que permetés jugar a billar francès (també anomenat de caramboles, era el de tota la vida al nostre país fins que el turisme i les modes anglosaxones el van anar desplaçant, substituint-lo pel *pool* i l'*snooker*), però en comptes de perfeccionar-lo, considerant les disipacions d'energia en els xocs i al llarg del recorregut de les boles (fins que totes quedessin en repòs) i incorporar-hi la possibilitat d'aplicar forces externes que no passessin pels centres de les boles, introduint-hi rotacions generadores dels anomenats "efectes", no es va anar més enllà de la hipòtesi mecànica de xoc elàstic perfecte (conservació de la quantitat de moviment) i d'un model *perpetuum mobile*, derivant cap a un joc que permetia fixar la posició i velocitat de les boles en un moment concret, i visualitzava les trajectòries determinades per aquests impulsos inicials i per la interferència mútua i amb les bandes elàstiques de la taula rectangular. Com que semblava poc elegant no acotar un final, forçant l'usuari a intervenir amb la tecla <Esc>, una de les entrades era precisament manifestar el nombre de xocs a tenir en compte. De tota manera, no seria difícil reconduir aquest material cap a un funcionament que reproduís el joc del billar: per simplificar, es podria considerar un moviment de les boles uniformement retardat, i el procés s'inscriuria en un bucle, sense fi (a avortar externament) o condicionat al comptatge de punts (punt = carambola ≡ ≡ doble xoc), on en cada iteració només s'assignés velocitat no nul·la a la bola jugadora. Això només és un suggeriment per al lector, perquè l'autor no modificarà res, més enllà de les mínimes i imprescindibles esmenes anunciades.

Abans de transcriure el codi, però, sí que caldria explicar en què consisteix la visualització esmentada. Per defecte es representen les trajectòries de totes les boles en joc, amb un cercle a la posició inicial i a cada posició de xoc, inclòs el cercle o parell de cercles corresponents a l'últim xoc (segons sigui bola-banda o bola-bola), mentre les altres trajectòries quedaran interrompudes sense cercle. (Cal tenir present que, allò que que en aparença és un únic tram de trajectòria, pot estar compost de diversos objectes-línia encadenats i alineats, i aleshores els punts de connexió indiquen on era la bola quan alguna de les altres xocava.) Alternativament, si contesteu afirmativament la pregunta *Vols efecte d'ANIMACIÓ (S/N)? <N>:*, hi ha la possibilitat de veure el moviment de les boles fins arribar a l'últim xoc programat. La velocitat de cada bola en la posició de sortida és assignada mitjançant sengles punts que, en relació a les respectives posicions, defineixen vectors. Si hem escollit l'efecte ANIMACIÓ, la traducció del vector a velocitat concreta en pantalla es pot graduar amb el valor assignat a la variable global **TVIS**. Naturalment, l'ajust a un determinat rang de velocitats perceptibles dependrà de la potència de l'equip i la representació de la bola: en el codi que us donem el contingut del bloc "BOLA" pot reduir-se a un trist cercle (i quedarà

d'allò més poc vistós, perquè només en veurem la circumferència) o complicar-se i ser un objecte-volandera (*arandela* o *donut*) sense forat (i ser vist com un cercle ple de color), segons que inutilitzem o no una línia de codi; doncs bé, la segona opció comporta un temps de regeneració un pèl més llarg, de manera que si canviem a la primera (regeneració un pèl més ràpida) i volem ser més papistes que el Papa també caldria reduir un pèl el valor de **TVIS** perquè l'escala vectorial fos igual.

En ambdós casos hi ha l'opció addicional a crear un document de text (.txt, tot i que l'usuari pot posar una altra extensió), amb tants registres (línies) com xocs bola-banda o bola-bola enregistrem, més un (el primer dóna la posició inicial de les boles). Cada registre es compon de **3 + 2n** camps (on **n** és el nombre de boles):

1 Temps acumulat des del moment inicial.
2 Temps transcorregut des del xoc precedent (des del moment inicial, en els dos primers registres).
3 Bola o boles implicades en el xoc, representades pel número d'ordre (si només n'hi ha una, és un xoc bola-banda).
.....
2+2N Coordenada **x** del centre de la bola **N** en el moment del xoc (l'afecti o no).
3+2N Coordenada **y** del centre de la bola **N** en el moment del xoc (l'afecti o no).
.....

Cal aclarir que un impacte múltiple (el xoc simultani d'una bola amb dues o més, o amb banda i bola o boles) es desglossarà en una seqüència d'impactes simples (xocs bola-bola i bola-banda, començant per l'impacte a la banda i seguint amb les boles implicades en l'ordre de numeració d'aquestes). L'autor suposa (ja no recorda què li passava pel cap fa setze anys, i no té ganes de posar-se a desxifrar anotacions breus, esquemes i desenvolupaments algebraics) que abordar-ho d'aquesta manera era força més fàcil i, malgrat el poc rigor teòric, l'arbitrarietat d'aquesta decisió potser responia més al factor d'imprevisibilitat que a la pràctica té aquesta mena d'impactes (no cal baixar a l'escala nanomètrica per entendre que la simultaneïtat en experiències com el joc de billar només es dóna aproximadament). Una incidència menor que es produeix en aquests casos és l'existència de trams de trajectòria de longitud nul·la amb la bola que causa doble impacte i que, si editem el fitxer de posicions, veurem reflectits en posicions consecutives idèntiques d'aquesta bola; la part negativa és l'aparició de missatges com *Línea de longitud cero creada en (4.4219, 7.7176, 0.0000)*, que la desactivació de **CMDECHO** no aconsegueix inhibir.

Per pura mandra, l'autor ha copiat el codi i l'ha deixat tal i com havia quedat setze anys enrera (tret dels ajustos anunciats), amb certs tics ara ja superats i que molt probablement eren la petjada recent d'haver-se passat cinc anys amb la Tesi Doctoral, programant en Pascal: a diferència d'aquell llenguatge compil·lat, amb AutoLISP no n'hi ha prou a incloure unes funcions en el cos de la definició d'una altra per aconseguir que, fora del temps d'execució de la funció-mare, les funcions-filles quedin descarregades (com passa amb les variables locals de cada funció, per exemple), sinó que cal eliminar-les explícitament. Ho podeu veure amb **BILLAR** (funció-mare) i **INICI** (funció filla), sense que calgui recórrer a **ATOMS-FAMILY**, fent primerament **(setq L (ATOMS-FAMILY 0))** per tenir la relació de tots els noms reservats en forma de llista, i després **(member 'INICI L)** per comprovar si la funció **INICI** està o no disponible; és molt més senzill escriure **!INICI** i mirar si el seu valor és del tipus **#<SUBR @02c310dc INICI>** o és **nil**. D'aquesta manera veureu com abans de carregar l'aplicació BILLAR.LSP, fent **(load "BILLAR")**, ni **BILLAR** ni **INICI** estan carregades, com és lògic; després de fer-ho, només haureu carregat **BILLAR**, però així que hagueu executat per primera vegada aquesta funció, fent **(BILLAR)**, les tindreu totes (**BILLAR**, **INICI** i demés filles de **BILLAR**). L'única manera de desprendre'ns de **INICI** i alliberar memòria serà fer **(set 'INICI nil)**, manualment o incloent aquesta sentència cap al final de la definició de **BILLAR**). Així que, des de fa molts anys l'autor té el costum de situar totes les funcions al mateix nivell i deixar-se de romanços, tret que l'arquitectura del programa ho demani (podreu comprovar-ho des del primer capítol), però aquest programa encara respon a l'anterior filosofia. Si voleu canviar-la, res més senzill, i el primer que recomanaria és canviar la forma **(defun BILLAR ...)** per **(defun C:BILLAR ...)** i d'aquesta manera us estalviareu parèntesis, invocant **BILLAR** com una ordre AutoCAD.

Inicialment es va crear una versió (BILLAR.LSP) pensada exclusivament per a tres boles:

```
(defun BILLAR (/ ZERO INFIN L TT N K KT O D F W WX WY VIS TVIS TMIN KMIN KMIN0
                 E1 E2 E3 O1 O2 O3 P1 P2 P3 V1 V2 V3)
```

```
(defun INC () (setq K (1+ K) KT (itoa K)))

(defun RS (L N) (read (strcat L N)))

(defun ERS (L N) (eval (RS L N)))

(defun INICI (/ P J OK)
  (setvar "CMDECHO" 0)
  (command "_UCSICON" "_OFF" "_UCS" "_W")
  (setq ZERO 1E-9 INFIN 1E+9 TVIS 0.1 TT 0 KMIN0 ""
        P1 (getpoint "\nTAULA DE BILLAR:\n  Primera cantonada <0.0,0.0>: ")
        P1 (if P1 P1 '(0.0 0.0))
        P (mapcar '+ P1 '(220.0 110.0))
        P2 (getcorner P1 (strcat "\n  Segona cantonada <"
                                 (rtos (car P)) "," (rtos (cadr P)) ">: "))
        P2 (if P2 P2 P)
        O (mapcar '/ (mapcar '+ P1 P2) '(2.0 2.0))
        W (mapcar 'abs (mapcar '- P1 P2))
        WX (/ (car W) 2.0) WY (/ (cadr W) 2.0))
  (command "_LAYER" "_OFF" "*" "_Y" "_N" "1,2,3" "_C" 7 "1" "_C" 2 "2" "_C" 1
                            "3" "_M" "W" "_C" 3 "" ""
           "_ZOOM" P1 P2 "_ZOOM" ".95X"
           "_ZOOM" "_C" O "" "_UCS" "_O" O
           "_PLINE" (list (- WX) (- WY)) (list WX (- WY))
                    (list WX WY) (list (- WX) WY) "C")
  (setq W (entlast))
  (prompt "\nBOLES:\n  Radi de les boles <3.0>: ")
  (command "COORDS" 2 "CIRCLERAD" 3 "_CIRCLE" '(0 0) PAUSE
         ; Quan convingui representar només el contorn de les boles, sense
         ; omplir-lo de color, caldrà inutilitzar les dues línies següents:
           "_ERASE" "_L" ""
           "FILLMODE" 1 "_DONUT" 0 (* 2 (getvar "CIRCLERAD")) "@" ""
           "_CHANGE" "_L" "" "_P" "_LA" "0" ""
           "_BLOCK" "BOLA" "@" (entlast) "")
  (setq D (getvar "CIRCLERAD") WX (- WX D) WY (- WY D) D (* 2 D) K 0)
  (repeat 3
    (INC)
    (command "_LAYER" "_S" KT "")
    (while (progn
             (prompt (strcat "\n  Situa BOLA " KT ": "))
             (command "_INSERT" "BOLA" "_S" 1 "_R" 0 PAUSE)
             (setq P (getvar "LASTPOINT") J 0 OK T)
             (if (or (> (abs (car P)) WX) (> (abs (cadr P)) WY))
                 (setq OK ())
                 (while (and OK (< (setq J (1+ J)) K))
                        (if (< (distance P (ERS "O" (itoa J))) D)
                            (setq OK ()))))
             (not OK))
           (prompt "\n    La ubicació de la bola NO ÉS CORRECTA!")
           (command "_U"))
    (set (RS "O" KT) (list (car P) (cadr P)))
    (set (RS "E" KT) (entget (entlast))))
  (setq L (cons (list TT 0 " " O1 O2 O3) L) K 0)
  (repeat 3
    (INC)
    (prompt (strcat "\n  Vector velocitat BOLA " KT " <en repòs>: "))
    (command "_UCS" "_O" (ERS "O" KT))
    (setq P (getpoint '(0.0 0.0)))
    (set (RS "V" KT) (if P P '(0.0 0.0)))
    (command "_UCS" "_P"))
  (setq N (getint "\n\nNombre de XOCS (bola-banda o bola-bola): "))
  (initget "Si No")
  (setq VIS (getkword "\n\nVols efecte d'ANIMACIÓ (S/N)? <N>: ")
        VIS (if (= VIS "Si") VIS)))

(defun ASSIGNA () (if (< TK TMIN) (setq TMIN TK KMIN TT)))
```

```
(defun T_A (O V TT / OX OY VX VY TX TY TK)
  (setq OX (car O) OY (cadr O)
        VX (car V) VY (cadr V)
        TX (if (equal VX 0 ZERO)
               INFIN
               (/ (- (if (> VX 0) WX (- WX)) OX) VX))
        TY (if (equal VY 0 ZERO)
               INFIN
               (/ (- (if (> VY 0) WY (- WY)) OY) VY))
        TK (min TX TY))
  (ASSIGNA))

(defun T_AB (OA OB VA VB TT / OX OY VX VY TX TY K1 K2 K3 K123 TK)
  (setq TK (cond ((= TT KMIN0)
                  INFIN)
                 ((progn
                    (setq OX (- (car OA) (car OB))
                          OY (- (cadr OA) (cadr OB))
                          VX (- (car VA) (car VB))
                          VY (- (cadr VA) (cadr VB)))
                    (or (and (>= OX D) (> VX 0))
                        (and (<= OX (- D)) (< VX 0))
                        (and (>= OY D) (> VY 0))
                        (and (<= OY (- D)) (< VY 0))))
                    INFIN)
                 ((equal (setq K1 (+ (expt VX 2) (expt VY 2))) 0 ZERO)
                  INFIN)
                 ((< (setq K2 (- (+ (* OX VX) (* OY VY)))) ZERO)
                  INFIN)
                 ((equal (setq K3 (- (+ (expt OX 2) (expt OY 2)) (expt D 2)))
                         0 ZERO)
                  0)
                 ((< (setq K123 (- (expt K2 2) (* K1 K3))) 0)
                  INFIN)
                 (T
                  (/ (- K2 (sqrt K123)) K1))))
  (ASSIGNA))

(defun O_P (O V PS) (set PS (mapcar '+ O (mapcar '* V (list TMIN TMIN)))))

(defun V_A (P VS / PX PY VX VY)
  (setq PX (car P) PY (cadr P)
        VX (car (eval VS)) VY (cadr (eval VS))
        VX (if (equal (abs PX) WX ZERO) (- VX) VX)
        VY (if (equal (abs PY) WY ZERO) (- VY) VY))
  (set VS (list VX VY)))

(defun NORM (V / VN)
  (setq VN (+ (* CANG (car V)) (* SANG (cadr V))))
  (list (* CANG VN) (* SANG VN)))

(defun TANG (V / VT)
  (setq VT (- (* SANG (car V)) (* CANG (cadr V))))
  (list (* SANG VT) (* (- CANG) VT)))

(defun V_AB (PA PB VAS VBS / ANG SANG CANG VA)
  (setq ANG (angle PA PB) SANG (sin ANG) CANG (cos ANG)
        VA (mapcar '+ (NORM (eval VBS)) (TANG (eval VAS))))
  (set VBS (mapcar '+ (NORM (eval VAS)) (TANG (eval VBS))))
  (set VAS VA))

(defun EXECUTA1 ()
  (setq K 0)
  (repeat 3
    (INC)
    (command "_LAYER" "_S" KT ""
             "_LINE" (ERS "O" KT) (ERS "P" KT) "")
```

```
    (if (or (= KT (substr KMIN 1 1)) (= KT (substr KMIN 2 1)))
        (command "_INSERT" "BOLA" "@" 1 1 0))))

(defun EXECUTA2 (/ NP NNPP S1 S2 S3)
  (setq NP (1+ (fix (/ TMIN TVIS)))
        NNPP (list NP NP)
        S1 (mapcar '/ (mapcar '- P1 O1) NNPP)
        S2 (mapcar '/ (mapcar '- P2 O2) NNPP)
        S3 (mapcar '/ (mapcar '- P3 O3) NNPP)
        O1 (mapcar '+ O O1)
        O2 (mapcar '+ O O2)
        O3 (mapcar '+ O O3))
  (repeat NP
    (entmod (subst (cons 10 (setq O1 (mapcar '+ O1 S1))) (assoc 10 E1) E1))
    (entmod (subst (cons 10 (setq O2 (mapcar '+ O2 S2))) (assoc 10 E2) E2))
    (entmod (subst (cons 10 (setq O3 (mapcar '+ O3 S3))) (assoc 10 E3) E3)))
  (if (= (strlen KMIN0) 1) (redraw W)))

(defun FITXER (/ NF RT P)
  (setq NF (open F "w"))
  (foreach E L
    (repeat (- 10 (strlen (setq RT (rtos (car E) 2 3)))) (princ " " NF))
    (princ RT NF)
    (repeat (- 8 (strlen (setq RT (rtos (cadr E) 2 3)))) (princ " " NF))
    (princ RT NF)
    (princ "  " NF)
    (princ (if (= (strlen (caddr E)) 1) (strcat (caddr E) " ") (caddr E)) NF)
    (setq K 2)
    (repeat 3
      (setq P (nth (setq K (1+ K)) E))
      (princ "  " NF)
      (foreach CP P
        (repeat (- 8 (strlen (setq RT (rtos CP 2 2)))) (princ " " NF))
        (princ RT NF)))
    (princ "\n" NF))
  (close NF))

(INICI) ; Inici de BILLAR
(repeat N
  (setq TMIN INFIN)
  (T_A O1 V1 "1")
  (T_A O2 V2 "2")
  (T_A O3 V3 "3")
  (T_AB O1 O2 V1 V2 "12")
  (T_AB O1 O3 V1 V3 "13")
  (T_AB O2 O3 V2 V3 "23")
  (O_P O1 V1 'P1)
  (O_P O2 V2 'P2)
  (O_P O3 V3 'P3)
  (if (= (strlen KMIN) 1)
      (V_A (ERS "P" KMIN) (RS "V" KMIN))
      (V_AB (ERS "P" (substr KMIN 1 1)) (ERS "P" (substr KMIN 2))
            (RS "V" (substr KMIN 1 1)) (RS "V" (substr KMIN 2))))
  (if VIS (EXECUTA2) (EXECUTA1))
  (setq L (cons (list (setq TT (+ TT TMIN)) TMIN KMIN P1 P2 P3) L)
        O1 P1 O2 P2 O3 P3 KMIN0 KMIN))
(setq L (reverse L)
      F (getstring "\n\nNom i extensió fitxer posicions <sense fitxer>: "))
(if (/= F "") (FITXER))
(command "COORDS" 1 "CMDECHO" 1)
(princ))
```

Després es va generalitzar per a **n** boles (VILLAR.LSP), amb 3 versions alternatives per a la funció **EXECUTA2**:

```
(defun VILLAR (/ ZERO INFIN TT NB NX K O* D F W WX WY VIS TVIS TM TMIN KMIN M N
               KMIN0 AA BB EE OO PP VV L)
```

```
(defun INICI (/ P P1 P2 J OK)
  (setvar "CMDECHO" 0)
  (command "_UCSICON" "_OFF" "_UCS" "_W")
  (setq ZERO 1E-9 INFIN 1E+9 TVIS 0.1 TT 0
        P1 (getpoint "\nTAULA DE BILLAR:\n  Primera cantonada <0.0,0.0>: ")
        P1 (if P1 P1 '(0.0 0.0))
        P (mapcar '+ P1 '(220.0 110.0))
        P2 (getcorner P1 (strcat "\n  Segona cantonada <"
                                 (rtos (car P)) "," (rtos (cadr P)) ">: "))
        P2 (if P2 P2 P)
        O* (mapcar '/ (mapcar '+ P1 P2) '(2.0 2.0))
        W (mapcar 'abs (mapcar '- P1 P2))
        WX (/ (car W) 2.0) WY (/ (cadr W) 2.0)
        NB (getint "\nBOLES:\n  Nombre de boles <3>: ")
        NB (if NB NB 3) K 0)
  (command "_LAYER" "_OFF" "*" "_Y" "")
  (repeat (min NB 6)
          (setq K (1+ K) J (if (= K 3) 7 K))
          (command "_LAYER" "_N" (itoa K) "_C" J (itoa K) ""))
  (command "_LAYER" "_M" "_W" "_C" 3 "" ""
           "_ZOOM" P1 P2 "_ZOOM" ".95X"
           "_ZOOM" "_C" O* "" "_UCS" "_O" O*
           "_PLINE" (list (- WX) (- WY)) (list WX (- WY))
                    (list WX WY) (list (- WX) WY) "C")
  (setq W (entlast))
  (prompt "\n  Radi de les boles <3.0>: ")
  (command "COORDS" 2 "CIRCLERAD" 3 "_CIRCLE" '(0 0) PAUSE
         ; Quan convingui representar només el contorn de les boles, sense
         ; omplir-lo de color, caldrà inutilitzar les dues línies següents:
           "_ERASE" "_L" ""
           "FILLMODE" 1 "_DONUT" 0 (* 2 (getvar "CIRCLERAD")) "@" ""
           "_CHANGE" "_L" "" "_P" "_LA" "0" ""
           "_BLOCK" "BOLA" "@" (entlast) "")
  (setq D (getvar "CIRCLERAD") WX (- WX D) WY (- WY D) D (* 2 D) K 0)
  (repeat NB (command "_LAYER" "_S" (itoa (1+ (rem K 6))) "")
             (setq J K)
             (repeat (- NB K 1)
                     (setq AA (append AA (list K))
                           BB (append BB (list (setq J (1+ J))))))
             (while (progn
                      (prompt (strcat "\n  Situa BOLA " (itoa (1+ K)) ": "))
                      (command "_INSERT" "BOLA" "_S" 1 "_R" 0 PAUSE)
                      (setq P (getvar "LASTPOINT") J -1 OK T)
                      (if (or (> (abs (car P)) WX) (> (abs (cadr P)) WY))
                          (setq OK ())
                          (while (and OK (< (setq J (1+ J)) K))
                                 (if (< (distance P (nth J OO)) D)
                                  (setq OK ()))))
                      (not OK))
                    (prompt "\n    La ubicació de la bola NO ES CORRECTA!")
                    (command "U"))
             (setq OO (append OO (list (list (car P) (cadr P))))
                   EE (append EE (list (entget (entlast))))
                   K (1+ K)))
  (setq L (cons (list TT 0 () OO) L) K 0)
  (foreach O OO
           (prompt (strcat "\n  Vector velocitat BOLA "
                           (itoa (setq K (1+ K))) " <en repòs>: "))
           (command "_UCS" "_O" O)
           (setq P (getpoint '(0.0 0.0))
                 VV (append VV (list (if P P '(0.0 0.0)))))
           (command "_UCS" "_P"))
  (setq NX (getint "\n\nNombre de XOCS (bola-banda o bola-bola): "))
  (initget "Si No")
  (setq VIS (getkword "\n\nVols efecte d'ANIMACIÓ (S/N)? <N>: ")
        VIS (if (= VIS "Si") VIS)))
```

```
(defun T_A (/ OX OY VX VY TX TY)
  (setq OX (car O) OY (cadr O)
        VX (car V) VY (cadr V)
        TX (if (equal VX 0 ZERO)
               INFIN
               (/ (- (if (> VX 0) WX (- WX)) OX) VX))
        TY (if (equal VY 0 ZERO)
               INFIN
               (/ (- (if (> VY 0) WY (- WY)) OY) VY)))
  (min TX TY))

(defun T_AB (/ OX OY VX VY TX TY K1 K2 K3 K123)
  (cond ((= TT KMIN0)
         INFIN)
        ((progn
           (setq OX (- (car OA) (car OB))
                 OY (- (cadr OA) (cadr OB))
                 VX (- (car VA) (car VB))
                 VY (- (cadr VA) (cadr VB)))
           (or (and (>= OX D) (> VX 0))
               (and (<= OX (- D)) (< VX 0))
               (and (>= OY D) (> VY 0))
               (and (<= OY (- D)) (< VY 0))))
           INFIN)
        ((equal (setq K1 (+ (expt VX 2) (expt VY 2))) 0 ZERO)
         INFIN)
        ((< (setq K2 (- (+ (* OX VX) (* OY VY)))) ZERO)
         INFIN)
        ((equal (setq K3 (- (+ (expt OX 2) (expt OY 2)) (expt D 2))) 0 ZERO)
         0)
        ((< (setq K123 (- (expt K2 2) (* K1 K3))) 0)
         INFIN)
        (T
         (/ (- K2 (sqrt K123)) K1))))

(defun O_P (O V) (mapcar '+ O (mapcar '* V (list TMIN TMIN))))

(defun COPIA () (setq WW (cons (nth (setq K (1+ K)) VV) WW)))

(defun V_A (P V / PX PY VX VY WW)
  (setq PX (car P) PY (cadr P)
        VX (car V) VY (cadr V)
        VX (if (equal (abs PX) WX ZERO) (- VX) VX)
        VY (if (equal (abs PY) WY ZERO) (- VY) VY) K -1)
  (repeat M (COPIA))
  (setq WW (cons (list VX VY) WW) K M)
  (repeat (- NB M 1) (COPIA))
  (setq VV (reverse WW)))

(defun NORM (V / VN)
  (setq VN (+ (* CANG (car V)) (* SANG (cadr V))))
  (list (* CANG VN) (* SANG VN)))

(defun TANG (V / VT)
  (setq VT (- (* SANG (car V)) (* CANG (cadr V))))
  (list (* SANG VT) (* (- CANG) VT)))

(defun V_AB (PA PB VA VB / ANG SANG CANG WW)
  (setq ANG (angle PA PB) SANG (sin ANG) CANG (cos ANG) K -1)
  (repeat M (COPIA))
  (setq WW (cons (mapcar '+ (NORM VB) (TANG VA)) WW) K M)
  (repeat (- N M 1) (COPIA))
  (setq WW (cons (mapcar '+ (NORM VA) (TANG VB)) WW) K N)
  (repeat (- NB N 1) (COPIA))
  (setq VV (reverse WW)))
```

```
  (defun EXECUTA1 ()
    (setq K -1)
    (mapcar '(lambda (O P)
               (setq K (1+ K))
               (command "_LAYER" "_S" (itoa (1+ K)) ""
                        "_LINE" O P "")
               (if (or (= K (car KMIN)) (= K (cadr KMIN)))
                   (command "_INSERT" "BOLA" "@" 1 1 0)))
             OO PP))

; La major lentitud de VILLAR.LSP amb 3 boles respecte a BILLAR.LSP (triga
; un 10% més) és deguda precisament en aquesta funció.
  (defun EXECUTA2 (/ N NN SS)
    (setq N (1+ (fix (/ TMIN TVIS))) NN (list N N)
          SS (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP)
          OO (mapcar '(lambda (O) (mapcar '+ O O*)) OO))
    (repeat N
            (setq OO (mapcar '(lambda (O S) (mapcar '+ O S)) OO SS))
            (mapcar '(lambda (O E) (entmod (subst (cons 10 O) (assoc 10 E) E)))
                    OO EE))
    (if (= (length KMIN0) 1) (redraw W)))

; ALTERNATIVA 1: una mica més ràpid (triga un 8% més que BILLAR.LSP)
  (defun 1XECUTA2 (/ N NN SS)
    (setq N (1+ (fix (/ TMIN TVIS)))
          NN (list N N)
          SS (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP)
          OO (mapcar '(lambda (O) (mapcar '+ O O*)) OO))
    (repeat N
      (setq OO (mapcar '(lambda (O S E)
                          (entmod (subst (cons 10 (setq O (mapcar '+ O S)))
                                         (assoc 10 E) E)) O)
                       OO SS EE)))
    (if (= (length KMIN0) 1) (redraw W)))

; ALTERNATIVA 2: una mica més ràpid (triga un 6% mès que BILLAR.LSP), però
; amb valors alts de NX pot donar desviacions, en calcular-se totes les
; posicions per via incremental a partir de la inicial.
  (defun 2XECUTA2 (/ N NN)
    (setq N (1+ (fix (/ TMIN TVIS)))
          NN (list N N)
          OO (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP))
    (repeat N (setq EE (mapcar '(lambda (O E)
                                  (entmod
                                    (subst (cons 10 (mapcar '+ (cdr (assoc 10 E))
                                                            O))
                                           (assoc 10 E)
                                           E)))
                               OO EE)))
    (if (= (length KMIN0) 1) (redraw W)))

  (defun FITXER (/ NF RT)
    (setq NF (open F "w"))
    (foreach E L
      (repeat (- 10 (strlen (setq RT (rtos (car E) 2 3)))) (princ " " NF))
      (princ RT NF)
      (repeat (- 8 (strlen (setq RT (rtos (cadr E) 2 3)))) (princ " " NF))
      (princ RT NF)
      (princ "  " NF)
      (setq RT (if (= (length (caddr E)) 0)
                   ""
                   (itoa (1+ (car (caddr E)))))
            RT (if (= (length (caddr E)) 2)
                   (strcat RT "-" (itoa (1+ (cadr (caddr E)))))
                   RT))
      (princ RT NF)
      (repeat (- 3 (strlen RT)) (princ " " NF))
```

```
    (foreach P (cadddr E)
      (princ "  " NF)
      (foreach CP P
        (repeat (- 8 (strlen (setq RT (rtos CP 2 2)))) (princ " " NF))
        (princ RT NF)))
    (princ "\n" NF))
  (close NF))

(INICI) ; Inici de VILLAR
(repeat NX
  (setq TMIN INFIN K -1)
  (mapcar '(lambda (O V)
             (setq K (1+ K) TM (T_A))
             (if (< TM TMIN) (setq TMIN TM KMIN (list K))))
          OO VV)
  (mapcar '(lambda (A B)
             (setq OA (nth A OO) OB (nth B OO)
                   VA (nth A VV) VB (nth B VV) TM (T_AB))
             (if (< TM TMIN) (setq TMIN TM KMIN (list A B))))
          AA BB)
  (setq PP (mapcar 'O_P OO VV) M (car KMIN) N (cadr KMIN))
  (if (= (length KMIN) 1)
      (V_A (nth M PP) (nth M VV))
      (V_AB (nth M PP) (nth N PP) (nth M VV) (nth N VV)))
  (if VIS (EXECUTA2) (EXECUTA1))
  (setq L (cons (list (setq TT (+ TT TMIN)) TMIN KMIN PP) L)
        OO PP KMIN0 KMIN))
(setq L (reverse L)
      F (getstring "\n\nNom i extensió fitxer posicions <sense fitxer>: "))
(if (/= F "") (FITXER))
(command "COORDS" 1 "CMDECHO" 1)
(princ))
```

Anem ara cap a les esmenes que haureu d'aplicar a BILLAR.LSP i/o VILLAR.LSP perquè funcionin. La primera potser no caldria suggerir-vos-la, perquè amb els comentaris que s'han fet abans a propòsit de **TVIS** segurament ja en teníeu prou. Però, a risc de resultar redundant (si no ho heu fet, no trigareu a adonar-vos-en que l'autor no acostuma a defugir aquest risc i que en cas de dubte s'estima més ser redundant i pesat com un plom que deixar-se coses en el tinter), ho farà. Aquesta variable en funcions de constant es va dimensionar fa setze anys, quan l'ordinador en què s'executava el programa era molt més lent, de forma que ara s'ha vist obligat a dividir per 10 aquell valor, després d'un breu tempteig, per recuperar el control de la velocitat de les boles sense haver de recórrer a uns vectors tan curts que posessin massa difícil fer punteria (orientar visualment el vector cap al lloc on volíem causar impacte). Així que la recomanació és: tot i ser el més probable que tingueu un equip més competitiu que el de l'autor, comenceu canviant la 4ª línia de la definició de funció **INICI**, deixant l'assignació **(setq ... TVIS 0.01 ...)** i, a partir d'aquest valor, aneu-lo reduint fins que la velocitat de les boles entri en un ordre de magnitud on us trobeu còmodes. La segona esmena també tè un caire empíric perquè, des de la versió 12 a les de referència aquí, 15 (ACAD 2000) i 16 (ACAD 2002 i ACAD 2004), a l'autor no li consta que hagin canviat les regles del joc de la funció **entmod**, ni pel que fa a la lletra menuda (funció complementària **entupd**, per forçar l'actualització en pantalla d'objectes subordinats de codi **-2** -com els atributs d'una inserció de bloc o els vèrtexs d'una polilínia-, sense fer una regeneració de tot el dibuix); tanmateix, havent reclamat l'efecte d'ANIMACIÓ, el mateix codi que fa setze anys li proporcionava aquesta il·lusió de moviment ja no respon: un cop aplicat el vector velocitat a l'última bola, tot esdevé immòbil fins que (transcorregut el temps en què se suposa que la taula és converteix en escenari d'un vodevil) les boles reapareixen sobtadament en les posicions finals. La causa del canvi se'ns escapa, però ha d'estar localitzada en aquests fragments:

```
(defun EXECUTA2 (/ NP NNPP S1 S2 S3) ; subordinada a funció BILLAR en BILLAR.LSP
  (setq NP (1+ (fix (/ TMIN TVIS))) NNPP (list NP NP)
        S1 (mapcar '/ (mapcar '- P1 O1) NNPP)
        S2 (mapcar '/ (mapcar '- P2 O2) NNPP)
        S3 (mapcar '/ (mapcar '- P3 O3) NNPP)
        O1 (mapcar '+ O O1) O2 (mapcar '+ O O2) O3 (mapcar '+ O O3))
```

```
    (repeat NP
      (setq O1 (mapcar '+ O1 S1) O2 (mapcar '+ O2 S2) O3 (mapcar '+ O3 S3))
      (entmod (subst (cons 10 O1) (assoc 10 E1) E1))
      (entmod (subst (cons 10 O2) (assoc 10 E2) E2))
      (entmod (subst (cons 10 O3) (assoc 10 E3) E3)))
    (if (= (strlen KMIN0) 1) (redraw W)))

  (defun EXECUTA2 (/ N NN SS)          ; subordinada a funció BILLAR en VILLAR.LSP
    (setq N (1+ (fix (/ TMIN TVIS))) NN (list N N)
          SS (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP)
          OO (mapcar '(lambda (O) (mapcar '+ O O*)) OO))
    (repeat N
            (setq OO (mapcar '(lambda (O S) (mapcar '+ O S)) OO SS))
            (mapcar '(lambda (O E) (entmod (subst (cons 10 O) (assoc 10 E) E)))
                    OO EE))
    (if (= (length KMIN0) 1) (redraw W)))
```

Si compareu les funcions **EXECUTA2** de BILLAR.LSP (en què hem alterat lleugerament la forma, però no el funcionament) i VILLAR.LSP, és fàcil adonar-se que, tret el nom de les variables (**NP**, **NNPP**, **(list S1 S2 S3)**, **(list O1 O2 O3)** i **(list P1 P2 P3)** de la primera corresponen a **N**, **NN**, **SS**, **OO** i **PP** de la segona) podríem dir que són calcades, i la llicència que ens hem permès modificant la primera era per fer-ne més evident la similitud: la diferència real és el nombre de boles, que són **3** a BILLAR.LSP i queda establert per l'usuari a VILLAR.LSP (és **(length EE)**). Doncs no sabem exactament per què, però a la versió 12 **entmod** no només anava canviant els punts d'ancoratge de les **NP ≡ N** insercions del bloc "BOLA" en la base de dades AutoCAD del dibuix, sinó que les seves representacions en pantalla es desplaçaven segons aquestes posicions (que és, ni més ni menys, allò que prometien i segueixen prometent els manuals d'AutoLISP). Sí que hi ha una cosa que ens crida l'atenció i és l'última sentència, **(if (= (length KMIN0) 1) (redraw W))**, sobre la necessitat de la qual ara només podem conjecturar (l'autor no és veu amb cor de recordar-ho, ni tampoc no té a mà l'AutoCAD 12 per provar-ho) que potser, després d'una sèrie ininterrompuda d'execucions d'**entmod** iniciada en un xoc bola-banda, la taula de billar (la polilínia rectangular, que paradoxalment és un objecte que no varia en tot el procés) s'esfumava misteriosament i perquè reaparegués calia redibuixar-la. És ben curiós que, quan es prova interactivament, no sols amb un cercle sinó amb la inserció d'un bloc constituït per només un cercle, dibuixant-los centrats a la posició 2,2, per exemple, i executant tot seguit les expressions

```
(setq E (entget (entlast)))
(entmod (subst '(10 8.0 6.0 0.0) (assoc 10 E) E))
```

els objectes es desplacen en pantalla a la posició 8,6, però quan intentem anar a un efecte d'animació i visualitzar posicions intermèdies, fent

```
(setq P '(2.0 2.0 0.0) E (entget (entlast)))
(repeat 10000
        (setq P (mapcar '+ P '(0.0006 0.0004 0.0)))
        (entmod (subst (cons 10 P) (assoc 10 E) E)))
```

torna a passar el mateix que en el programa: la immobilitat més aclaparadora uns segons i, de cop, el salt de la bola a 8,6. En canvi, si hi afegim un petit detall

```
(setq P '(2.0 2.0 0.0) E (entget (entlast)))
(repeat 10000
        (setq P (mapcar '+ P '(0.0006 0.0004 0.0)))
        (entmod (subst (cons 10 P) (assoc 10 E) E))
        (redraw))
```

s'opera el miracle de l'animació. Però, ATENCIÓ!: fixeu-vos que ens hem limitat a executar **(redraw)**; si, filant més prim, haguéssim fet **(redraw (entlast))**, creient que així evitaríem alentir innecessàriament el mòbil en el cas que la prova hagués estat realitzada en mig d'una sessió de dibuix (en què no calia que el redibuixat s'estengués a la resta d'objectes, eventualment molt nombrosos), hauríem begut oli perquè un altre cop hauríem perdut l'efecte d'animació. Com que això, ultra ser desconcertant, ja fa pujar la mosca al nas, l'autor se sent obligat a aventurar una hipòtesi, a risc que algun dia un autèntic expert en AutoCAD declari que els misteri tenia altres orígens. Descartat que la causa fos l'anul·lació efectiva de la primera pregunta de l'ordre **RESVISTA** (*¿Desea zooms rápidos? [Sí/No] <S>:*), que activa l'anomenada "pantalla virtual" (àrea en què els canvis de visualització no requereixen la regeneració del dibuix), i que si es manté a partir d'AutoCAD 2004 sols és per raons de compatibilitat amb versions anteriors, perquè el comportament descrit és comú amb l'AutoCAD 2000 i es dóna tant si contestem que sí com que no,

tanmateix tot apunta a que el problema ha d'estar relacionat amb els dos formats numèrics que usa AutoCAD: nombres reals (operacions lentes) per representar la geometria dels objectes a la base de dades del dibuix; nombres enters (operacions més ràpides) per representar coordenades de pantalla. És pràcticament impossible trobar versions AutoCAD de fa més de 15 anys (i disposar de sistemes operatius que les suportin, és clar), però sortosament no passa el mateix amb els llibres-paper: per exemple, a la sèrie de títols *AUTOCAD AVANZADO*, de J. López i J.A. Tajadura (McGraw-Hill, 1993), la primera referència a l'opció **vMáx** de l'ordre **ZOOM** i a una pantalla virtual pautada en enters ("*Para saber qué parte del dibujo se visualiza en pantalla y con qué precisión, AutoCAD trabaja con una pantalla virtual de más de cuatro mil millones de puntos en cada eje. La posición de cada punto en esta pantalla virtual son coordenadas con valores enteros. AutoCAD, mediante un proceso llamado regeneración del dibujo, transforma las coordenadas de su base de datos en coma flotante, al formato entero de la pantalla virtual.*") corresponen a la versió 12, per bé que a la versió 11 (1991) ja es parli de l'opció **Dinàmico** de la mateixa ordre i de l'ordre **RESVISTA**. D'altra banda, els enters de les primeres versions no donaven la talla (**–32.768 ≤ n ≤ 32.767**) i l'esmentada prestació no va ser possible fins que no en van ampliar el rang (**–2.147.483.648 ≤ n ≤ 2.147.483.647**). Si el pas es va donar a la versió 12, ¿no podria ser que la programació en AutoLISP encara trigués en adaptar-se al canvi, i que la versió utilitzada en el programa de 1994 treballés exclusivament en aritmètica real (coma flotant)? O, quan es començava a veure que la política comercial d'Autodesk, Inc. (que ja no tenia competidors que li fessin ombra en el CAD de propòsit general) s'orientava cada cop més a poder treure en el mercat una "nova" versió d'AutoCAD cada dos anys (a partir de 2003, seria cada any), a base de dosificar amb comptagotes els avenços que ja tenien en cartera, ¿no seria encara més plausible que allò que s'anunciava a la versió 12 no s'incorporés plenament fins la versió 13 (1995)? Seguint la mateixa pista de la col·lecció *AUTOCAD AVANZADO*, trobarem detalls significatius que avalen la teoria:

- Malgrat la cita entre parèntesi d'unes línies enrera, en el llibre corresponent a la 12 encara es presenta l'ordre **REDIBUJA** igual que en les versions anteriors: "*Esta orden permite limpiar la pantalla de las marcas auxiliares generadas en la creación del dibujo o en la designación de entidades*".
- Però en el corresponent a la versió 13, llegim: "*Esta orden actualiza la vista en el área gráfica del dibujo (pantalla real), a partir de la pantalla virtual con la que trabaja AutoCAD. Su primer efecto más visible es limpiar las marcas auxiliares generadas en la creación del dibujo [...]*".

Vet aquí, doncs, el que molt probablement encara passava quan el programa estava a càrrec de la versió 12: no s'havia implementat la representació paral·lela de les coordenades en valors enters; o, si ja s'havia implementat, la funció **entmod** no se n'aprofitava i després de cada avaluació es produïa una regeneració (almenys, si la variable de sistema **REGENMODE** seguia en l'estat per defecte, activada), cosa que garantia l'actualització visual de l'objecte modificat, tot i que per la via lenta. A partir de la versió 13, el canvi **entmod** de coordenades expressades en reals ha de comportar una traducció simultània a coordenades enteres de pantalla virtual, condició *sine qua non* de la seva manifestació gràfica, però totes les experiències realitzades apunten en la mateixa direcció: aquesta traducció només té lloc en l'Editor de Dibuix; si **entmod** se situa en un bucle, només en sortir-ne i acabar-se l'avaluació de **(BILLAR)** (en realitat, una mica abans: en executar-se **(command "COORDS" 1 "CMDECHO" 1)**) s'apreciaran les modificacions en pantalla o, per ser més exactes, s'apreciarà l'última modificació. Sortosament, hi ha una via alternativa que pot recuperar visualment les modificacions intermèdies, perquè no cal passar definitivament el control a l'Editor de Dibuix: només que li passem temporalment, mitjançant la funció **command** (o mitjançant **(redraw)**, que al cap i a la fi equival a fer **(command "_REDRAW")**), ja n'hi ha prou. Convé puntualitzar que ni els arxius d'ajut d'Autodesk, cada cop més telegràfics (probablement perquè hi ha més material a comentar) ni les publicacions sota llicència, són especialment aclaridors al respecte. En concret, a l'últim títol adquirit per l'autor sobre AutoLISP, *Programación con AutoCAD.- Personalización, AutoLISP y Visual Basic, desde la v.14 en adelante* (McGraw-Hill, 1999) dels abans esmentats J. López i J.A. Tajadura, secundats per B. Manso, tampoc no se'n treu l'aigua clara, perquè no és a l'article **entmod** (on simplement es comenta que la funció "*... recibe la lista de la entidad (obtenida con ENTGET y modificada después) y la inserta sustituyendo a la que existía [en la base de datos]*", sense parlar de visió en pantalla) sinó en el següent, relatiu a **entupd**, on es diu, com de passada, "*... Si se trata de una entidad principal, ENTMOD regenera directamente la entidad y ésta se visualiza en pantalla*" i segueix parlant de limitacions i restriccions, però només en relació al tipus d'objectes; no pas pel que fa a normes l'ús de la funció en el marc d'una estratègia de programació. Perdoneu, però algú ho havia de dir (Joan Tardà *dixit*).

En definitiva, si després de dibuixar el cercle (o la inserció del bloc constituït per un cercle) centrat a 2,2 i de desactivar la variable de sistema **CMDECHO**, fem

```
(repeat 10000 (command "_MOVE" "_L" "" "0.0006,0.0004" ""))
```

també aconseguirem efecte d'animació però més lent, com correspon a l'ús d'ordres AutoCAD mitjançant **command** (hauríem de substituir els valors explícits per **1000** i **"0.006,0.004"**, per compensar-ho), perquè en el context de l'Editor de Dibuix sí que l'actualització en pantalla és automàtica. De fet, podem accedir-hi utilitzant **command** per a qualsevol poca-soltada que no freni massa el desplaçament, com per exemple, consultar el valor actual d'una variable de sistema, i així veuríem que substituint en la línia final **(redraw)** per **(command "CMDECHO" "")**, en l'expressió de l'última pàgina, el resultat (tret d'haver minorat la velocitat) és el mateix:

```
(setq P '(2.0 2.0 0.0) E (entget (entlast)))
(repeat 10000 (setq P (mapcar '+ P '(0.0006 0.0004 0.0)))
              (entmod (subst (cons 10 P) (assoc 10 E) E))
              (command "CMDECHO" ""))
```

Malgrat tot, recórrer a **(redraw)** sembla més seriós (i és el dispositiu més ràpid; després ja modularem a la nostra conveniència les velocitats, amb **TVIS**). Doncs bé, sense saber del cert si l'origen del mal era tota aquesta història de la pantalla virtual i la traducció de valors reals a enters, és clar que la medecina **(redraw)** guareix, i l'aplicarem al nostre programa (si la Medicina recorre fonamentalment al mètode empíric, ¿per què aquí no podríem fer-ho?): **EXECUTA2** quedarà en la forma

```
(defun EXECUTA2 (/ NP NNPP S1 S2 S3)
  (setq NP (1+ (fix (/ TMIN TVIS))) NNPP (list NP NP)
        S1 (mapcar '/ (mapcar '- P1 O1) NNPP)
        S2 (mapcar '/ (mapcar '- P2 O2) NNPP)
        S3 (mapcar '/ (mapcar '- P3 O3) NNPP)
        O1 (mapcar '+ O O1) O2 (mapcar '+ O O2) O3 (mapcar '+ O O3))
  (repeat NP
    (entmod (subst (cons 10 (setq O1 (mapcar '+ O1 S1))) (assoc 10 E1) E1))
    (entmod (subst (cons 10 (setq O2 (mapcar '+ O2 S2))) (assoc 10 E2) E2))
    (entmod (subst (cons 10 (setq O3 (mapcar '+ O3 S3))) (assoc 10 E3) E3))
    (redraw)))
```

a BILLAR.LSP, i pel que fa a VILLAR.LSP (en què tenim tres versions alternatives),

```
(defun EXECUTA2 (/ N NN SS)
  (setq N (1+ (fix (/ TMIN TVIS))) NN (list N N)
        SS (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP)
        OO (mapcar '(lambda (O) (mapcar '+ O O*)) OO))
  (repeat N (setq OO (mapcar '(lambda (O S) (mapcar '+ O S)) OO SS))
            (mapcar '(lambda (O E) (entmod (subst (cons 10 O) (assoc 10 E) E)))
                    OO EE))
            (redraw)))

(defun 1XECUTA2 (/ N NN SS)
  (setq N (1+ (fix (/ TMIN TVIS))) NN (list N N)
        SS (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP)
        OO (mapcar '(lambda (O) (mapcar '+ O O*)) OO))
  (repeat N
    (setq OO (mapcar '(lambda (O S E)
                        (entmod (subst (cons 10 (setq O (mapcar '+ O S)))
                                       (assoc 10 E) E)) O)
                     OO SS EE)))
    (redraw)))

(defun 2XECUTA2 (/ N NN)
  (setq N (1+ (fix (/ TMIN TVIS))) NN (list N N)
        OO (mapcar '(lambda (O P) (mapcar '/ (mapcar '- P O) NN)) OO PP))
  (repeat N
    (setq EE (mapcar '(lambda (O E)
                        (entmod (subst (cons 10 (mapcar '+ (cdr (assoc 10 E))O))
                                       (assoc 10 E) E)))
                     OO EE))
    (redraw)))
```

Vegeu que en totes les versions es prescindeix (havent-nos quedat amb les ganes de saber si amb AutoCAD 12 era realment necessària o un residu pendent de depuració) de la sentència final **(if (= (length KMIN0) 1) (redraw W))**, perque **(redraw)** afecta ara tots els objectes que componen el dibuix, taula de billar inclosa.

A l'autor li agradaria dedicar aquest treball als seus col·legues Esteban Zorrilla i Javier Muniozguren, fecunds col·laboradors a la *Euskal Herriko Unibertsitatea* (Universitat del País Vasc). Tot i només haver tingut amb ells quatre i dos breus contactes incidentals, respectivament, podria dir que han orientat la seva manera d'integrar-se en el món universitari (diametralment oposada a la d'ells, model de carrera docent) i de lluitar per obrir camins a tot allò que es pogués catalogar com a recerca en un sentit ampli, en el context de l'àrea de coneixement Expressió Gràfica a l'Enginyeria (en el fons, no tan oposada). Del primer, que malauradament ja no hi és entre nosaltres i de qui, abans de tenir l'oportunitat de conèixer-lo més a fons, el corprenia l'enigmàtic somriure amb què t'escoltava (i que situaria entre el de l'ex-*lehendakari* Carlos Garaikoetxea i el del monstre Amon Goeth, en la magistral interpretació a càrrec de Ralph Fiennes a "La llista de Schindler"), recorda com tot un any i mig el va tenir per un inquisidor despietat i principal coadjuvant al desastre en què va acabar el primer Concurs a una plaça de Titular d'Universitat a què l'autor es va presentar (fixeu-vos que ha escrit coadjuvant i no responsable del desastre: el primer responsable va ser el propi aspirant, en no voler assabentar-se de quin era el joc que es jugava). Passat el segon i definitiu Concurs en què, casualment, el Dr. Zorrilla tornava a formar part del Tribunal, va saber que, en el primer, aquest i els altres membres del Tribunal només s'havien limitat a secundar el President (teòric millor valedor d'un candidat que, amb la seva actuació maldestre, el va obligar a fer honor a la fama de jutge implacable). Les altres dues ocasions en què van coincidir, ara ja jugant al mateix bàndol, van ser igualment concursos, i al marge de l'actuació estrictament acadèmica va sorgir un cert corrent de simpatia. L'autor va descobrir en l'Esteban Zorrilla un docent enamorat de la seva professió i una persona dotada d'un subtil i envejable sentit de l'humor. Va ser ell l'únic de qui recorda que li parlés amb autèntic entusiasme d'uns Congressos d'Enginyeria Gràfica anuals que ja estaven en marxa, dient que de moment era el que hi havia i que l'important era participar; que d'entre un munt d'iniciatives discretes de tant en tant n'excel·lia alguna de realment interessant i que si no s'hi posava tothom, sense complexos, la nostra àrea de coneixement mai no arribaria quantitativament ni qualitativa a la producció tecnològica d'altres. Llàstima que poc després, entre la desafortunada actuació del llavors Director del Departament, negant suport real a una iniciativa dels senyors J i K, i l'acció de l'E.T.S.E.T.B. desterrant-los junt amb la Sra. L al Campus Sud, la desmoralització de l'autor l'allunyés d'aquesta actitud i el portés a refugiar-se en realitzacions com la inclosa línies enrera a títol de propina. El present treball no hi difereix massa pel que fa a tarannà, però ja s'ha explicat en l'inici d'aquesta *Introducció* que l'autor mai no s'hi hauria ficat si arriba a sospitar que duria tant de feina.

Al segon, amb qui no comparteix ni aficions cinegètiques ni l'amor a la velocitat motoritzada, li ha d'estar sincerament agraït pel capot amb què va intentar tapar el ridícul espantós en què l'autor va caure quan, en el torn de preguntes de la 2ª prova d'un Concurs a la Universitat Jaume I (Castelló, novembre de 1999), en lloc de declinar l'honor perquè del tema (*Tolerancias geométricas*) ni en sabia ni volia saber-ne res, es va sentir obligat a improvisar una objecció que el va deixar amb el cul a l'aire (no a la doctoranda, sinó a l'interpel·lador): d'aquella galdosa intervenció se'n desprenia que, quan no s'explicita cap tolerància, la tolerància és nul·la! Ni el mal moment emocional que estava travessant ni l'afecció gripal (en realitat, el diagnòstic una setmana més tard va ser citomegalovirus) que va contreure a l'hotel, pel nul manteniment de les habitacions des de la temporada d'estiu, justificaven tanta estupidesa. Una altra experiència que l'autor recorda d'aquella estada a Castelló va ser el descobriment d'un dels tics més curiosos de la dreta valenciana (que després, en freqüentar sovint la ciutat de València per raons familiars, ha tingut ocasió de reviure sense arribar a acostumar-s'hi mai): l'obstinació amb què molts valencians a qui un aborda quan estan mantenint una conversa en valencià, es passen al castellà quan s'adonen que el nou interlocutor prové de més enllà de Vinaròs, se suposa que per allò que valencià i català són dues llengües que no tenen res a veure. En aquest cas es tractava d'un catedràtic de la U.P.V., de qui no citarà el nom, membre del Tribunal de Tesi i amb qui va tenir ocasió de xerrar distesament una tarda a l'antiga Estació de Castellò, tot esperant el tren (ell anava a Vila-real, on vivia, i l'autor a València, a veure la que seria la seva dona): d'una amabilitat esquisida, no va haver-hi manera que

li dirigís una sola paraula en valencià. Però tornem a la persona de qui parlava, el president d'aquell Tribunal. Perquè, quan sis anys i mig després l'autor se'n va recordar dels consells de l'Esteban Zorrilla i va decidir presentar al *XVIII Congreso Internacional de Ingeniería Gráfica*, aprofitant que es faria a Sitges, tres comunicacions (*Notas sobre la orden AJUSTARIMG de AutoCAD: acción de BRILLO, CONTRASTE i DIFUMINADO*; *AutoCAD: adaptación de un bloque ortonormal 2D [3D] a un paralelógramo [paralelepípedo] dado*, i *Sinópsis y notas de trabajo sobre AutoCAD: inventario y propuestas*) que, en realitat eren l'embrió de tres treballs, es va trobar amb el rebuig de totes tres per part de'una comissió de selecció (*Comité Científico*) presidida per Javier Muniozguren. De primer, li va doldre pensar que el president pogués haver influït en les decisions, a causa d'una pèssima opinió sobre el ponent forjada en Castelló. Després, però, va tenir motius suficients per creure que la seva intervenció difícilment podia haver anat més enllà d'una tasca de coordinació, d'unificació de criteris o, en última instància, de resolució dels casos dubtosos, i que la responsabilitat directa del dictamen era dels dos vocals instats a pronunciar-se sobre cada comunicació rebuda. Si aquesta coincidència en la persona d'en Javier Muniozguren s'ha portat a col·lació només és perquè això li dóna peu a l'autor per posar fi a aquesta *Introducció* amb un altre pas de rosca: la primera comunicació (qualificada amb 1,26 punts sobre 5) conté un dels elements instrumentals amb què l'autor compta per "... reprendre un treball de més volada que havia interromput vuit anys enrera", segons explicava a la pàgina 2; la segona (qualificada amb 1,85 punts sobre 5) s'ha acabat materialitzant amb el document de UPCommons > E-prints UPC *Manipulació geomètrica dels blocs d'AutoCAD*, citat a la pàgina 4 (http://upcommons.upc.edu/e-prints/handle/2117/1603); pel que fa a la tercera, paradoxalment la "més ben qualificada" (amb 2,46 punts sobre 5!), mai no es concretarà en cap treball, perquè l'exhort amb què l'autor en rematava el resum (*"Incluyo la invitación a colaborar, por puro placer, en el diseño de un simulador de reactivo posicional de cursor"*) i que es concretava en les tres últimes pàgines (*3. Invitación a modo de conclusiones*) no va tenir cap resposta, malgrat haver-se distribuït en el CD entregat als participants al XVIII Congrés i ser accesible en una web activa ben bé dos anys. Actualment, els accessos a les comunicacions són independents i *Sinópsis y notas de trabajo sobre AutoCAD: inventario y propuestas* es pot trobar a http://www.ingegraf.es/XVIII/PDF/Comunicacion17044.pdf. L'autor no es fa massa il·lusions, malgrat haver deixat aquest enllaç com qui no vol la cosa: la carrera docent, sota el pretext d'assolir el més alt nivell (conseqüentment amb la promesa d'abans, s'ha eludit el mot que excel·leix a tots els papers oficials), s'ha complicat de tal manera que els aspirants a funcionari ja no poden permetre's frivolitats i els funcionaris malden per aconseguir el màxim de punts REC i TRANS. Potser recollirà el guant algun jubilat? Quan la recerca ja no està determinada (més que condicionada) per l'afany d'engreixar el currículum o d'incrementar els ingressos, ¿recuperarà, amb una legítima motivació lúdica, la seva independència? ¿O tot això són jeremiades indignes d'algú que ha tingut responsabilitats docents i que probablement ja rapapieja, perquè fora del Mercat res no té sentit?

L'autor reservava per a una futura publicació sobre conversió, compactació i ajust d'imatges en AutoCAD (implícita a la cita de la pàgina 2 "... reprendre un treball de més volada que havia interromput vuit anys enrera", repetida línies amunt) una dedicatòria encara més entranyable, però a l'hora de posar punt i final a aquesta *Introducció* s'ho ha pensat millor: allò de *no hay dos sin tres* podria no acomplir-se en el cas dels treballs publicats a *UPCommons > E-prints UPC*, sigui perquè la parca no hagués donat prou de temps o perquè algunes de les opinions expressades en aquest segon haguessin incomodat a algú amb prou influència com per impedir-ho, així que fóra bo assegurar-se la jugada i fer-la ara. Unes pàgines enrera, abans de deixar-se endur per la rancúnia amb què recorda el desterrament al Campus Sud, l'autor ha reconegut que "l'esperit d'obertura i cooperació interdisciplinar de l'E.T.S.E.T.B. va suposar una alenada d'aire fresc" i que "als seus companys se'ls veia bona gent i en aquella escola hi havia possibilitat de fer coses noves", però pecaria d'omissió si es quedés aquí, perquè qui ja era un *carroza* que tornava del fred va tenir el privilegi de conèixer un grapat de professors brillants (i, per damunt de tot, excel·lents persones), gairebé tots més joves (tret d'un de la seva mateixa lleva i que a hores d'ara deu estar al Perú, fent bona feina, com sempre) i del Departament d'Enginyeria Electrònica (amb el temps, la majoria va romandre a la U.P.C. i la resta se'n van anar a la Universitat Rovira i Virgili). Sempre els recordarà com el màxim exponent de solidaritat que ha conegut. Ells saben perquè.

(Aquesta *Introducció* va ser escrita entre març i maig de 2010. Com que l'autor no se'n penedeix ni d'una coma però els quasi tres anys següents han estat mogudets, us remet a la p. 495 on, poc abans d'acabar, ha tingut ocasió de posar-la al dia.)

Inventariar les solucions: un cul-de-sac

Abans d'entrar en matèria, l'autor vol fer un advertiment a propòsit del caràcter, més que sinuós, erràtic, del discurs amb què el lector es trobarà. La sensació de veure-se-les amb una manualitat *patchwork* més que amb un assèptic text expositiu acceptablement estructurat, segurament té molt a veure amb la desgana amb què s'ha redactat, circumstància que a la vegada és conseqüència de l'allargassament d'unes previsions inicials massa optimistes, comentades a l'inici de la *Introducció*, però també de la manera peculiar de treballar que té l'autor: incapaç de recordar quina era la situació que dues setmanes enrera l'havia dut a deixar a mitges allò que es portava entre mans i a explorar en una nova direcció, ni es plantejava esperar a culminar la feina de programació per posar-se a escriure'n el text explicatiu, per l'enorme dificultat de rememorar el procés des del començament, basant-se només en el contingut dels successius arxius .lsp, i per la mandra que li produïa pensar en reviure'l sense l'estímul d'explorar un territori verge; però incapaç també d'anar emplenant en paral·lel un quadern, amb les anotacions pulcres i ordenades amb què ho feien, per exemple, el Sr. J i la Sra. L de la *Introducció*, no ha trobat cap altre sistema per sortir-se'n que anar alternant avenços parcials en programació amb el text que ha de narrar el procés i justificar-ne els passos; quan no ho fa així encara és pitjor, perquè a l'hora de recordar el perquè d'algunes decisions preses, molt sovint s'embolica en raonaments que en realitat mai no havia efectuat (i que força vegades, ¡oh misèries de la condició humana!, l'han dut a descobrir fallades). La cosa no tindria més transcendència si no fos perquè la progressió mai no ha estat lineal sinó accidentada i plena de rectificacions que, en intentar traslladar al text explicatiu, encara el fan més abstrús (i sovint contradictori, malgrat els recursos retòrics esmerçats en maquillar els pegats). També és veritat que ja ens hi havíem referit, a la falta de sincronia entre l'ordre per capítols i la trajectòria real del treball, que esquematitzàvem amb una *M* ciríl·lica, però si teniu en compte que aquesta primera al·lusió l'autor la feia en la *Introducció* que va redactar en un moment en què ingènuament creia que ja s'acostava la recta final (al cap de dos anys i mig de pensar en ficar cullerada en això dels sudokus) i que aquestes línies les escriu dos anys després (a sis mesos de donar-ho per acabat, perquè s'ha compromès a no hi dedicar-hi més de cinc anys), ja podeu imaginar que mentrestant les incidències s'han multiplicat i les complicacions han augmentat el divorci entre exposició en negre sobre blanc i l'evolució real del codi AutoLISP.

En algun lloc, més endavant, l'autor ha deixat anotat "això no és un llibre seriós sinó un quadern de bitàcola". Com acaba de confessar, en ser incapaç de dur-ne un en paral·lel (també per mandra, que tot s'ha de dir) ha intentat matar dos ocells d'un tret: el text expositiu havia de ser una mena de recordatori per a ell mateix i una font d'informació entenedora per al lector. Però, com que cadascun d'aquests objectius té requeriments específics i no sempre compatibles, el més probable és que la solució de compromís adoptada sigui poc satisfactòria en tots dos nivells: efectivament, de cara al lector s'haurien hagut d'obviar moltes incidències, i amb això hagués millorat tant la qualitat pedagògica del text com la pròpia imatge de l'autor (en les publicacions científicotècniques totes les troballes venen rodades i gairebé mai no es reconeixen les ensopegades ni la intervenció de l'atzar); pel que fa a aquest, li hagues anat millor mantenir un to estrictament narratiu, però va redactar massa aviat un *ÍNDEX* que teòricament l'havia d'ajudar a estructurar el treball (encara no hi figurava l'*Epíleg*, que ha estat un recurs d'última hora) i, quan han anat sorgint les dificultats de debò, ja no s'ha atrevit a desmuntar-lo. Tot i que l'autor s'ha esforçat a esmorteir dissonàncies ampliant l'orquestració, sovint es pot reconèixer si un paràgraf reprodueix allò que va ser però ha calgut esmenar, allò en què l'autor treballava quan el va escriure o allò en què confia però encara no ha verificat experimentalment, en l'ús d'uns temps verbals (passat, present o futur) que, si a més es barregen, donen certa sensació de desmanegament.

Ens hem prodigat en tantes explicacions prèvies perquè aquest primer capítol és un clar exemple del que s'ha dit. D'entrada, la seva ubicació respecta la cronologia, perquè realment la pretensió de construir i emmagatzemar totes les solucions és el primer objectiu que l'autor es va fixar, quan la intuició sobre la possibilitat de fer sudokus a partir d'un catàleg de solucions el va animar a posar-se a treballar en aquesta línia. També el seu títol, perquè després de dissenyar uns instruments i d'haver-se dedicat a inventariar totes les combinacions possibles de les xifres 1 al 9, sense repetició (files), i unes quantes tríades de files que mantenien les

regles establertes per als sudokus (no repeticions, en cap ni un dels nou tercets verticals ni dels tres subconjunts de 9 valors definits pels tercets 1 a 3, 4 a 6 i 7 a 9), ja no va arribar a construir cap solució entesa com a tríada de tríades compatibles (en el sentit de no haver-hi repeticions en cap de les nou columnes), perquè a partir de l'ocupació en memòria dels arxius de tríades i del seu temps de gestació, ja s'adonava que la pretensió de disposar de totes les solucions era un cul-de-sac. Tanmateix, durant quatre anys i mig el present capítol s'ha quedat en blanc (o, per a ser exactes, reduït a un títol i a una idea prou definida de quin havia de ser el contingut), mentre l'autor desenvolupava, ara caic, ara m'aixeco, els altres tretze, i ha estat justament quan ja els tenia tots almenys encarrilats que ha decidit posar-s'hi (potser perquè, havent quedat sense excuses per seguir-lo postergant, ja tocava). Per acabar-ho d'adobar, quan ha repassat les funcions que havia preparat per crear els arxius de tríades i ha inspeccionat el contingut d'aquests arxius, s'ha adonat que totes dues coses eren manifestament millorables: de mitjana, la grandària dels arxius es podia reduir 40 vegades, i la durada de la seva creació, 100; però, malgrat l'optimització d'aquests recursos intermedis, els objectius estratègics (inventariar totes les solucions) seguien sent inabastables. Així que el lector no s'ha d'estranyar que, si més no en aquest capítol, adoptem el punt de vista del narrador omniscient i ens referim amb tota naturalitat a la patinada inicial (en el doble sentit de plantejament utòpic i de realització poc reeixida), a la millora d'algunes eines i a la possibilitat d'aprofitar-ne algunes com a complement del programa que es desenvoluparà a partir del capítol següent.
Una última qüestió prèvia consistirà a posar-nos d'acord sobre la terminologia que adoptarem al llarg de tota l'obra. Anomenarem:

- Escaquer 9×9, al recinte format per la juxtaposició de 81 caselles quadrades que s'arrengleren formant 9 files (numerades de baix a dalt) i 9 columnes (numerades d'esquerra a dreta). Les caselles poden estar buides o contenir un nombre enter de l'**1** al **9**, que anomenarem valor de la casella.
- Requadre 9×3, tríada de files, tríada horitzontal (o simplement tríada), al conjunt de les files 1, 2 i 3, 4, 5 i 6 o 7, 8 i 9 (designats com a inferior, central o superior, o també numerats de baix a dalt).
- Requadre 3×9, tríada de columnes o tríada vertical, al conjunt de les columnes 1, 2 i 3, 4, 5 i 6 o 7, 8 i 9 (designats com a esquerre, central o dret, o també numerats d'esquerra a dreta).
- Requadre 3×3, a la intersecció d'un requadre 9×3 amb un de 3×9 (designats com a inferior-esquerre, inferior-central, inferior-dret, central-esquerre, central, central-dret, superior-esquerre, superior-central i superior-dret).
- SUDOKU-SOLUCIÓ (o simplement solució), a un escaquer 9×9 amb totes les caselles plenes, però de manera que cap fila, cap columna ni cap requadre 3×3 no tinguin valors repetits (o el seu equivalent: que tinguin tots els valors, de l'**1** al **9**).
- SUDOKU-PROBLEMA (o simplement sudoku), a una SUDOKU-SOLUCIÓ incompleta (queden caselles per emplenar) en què només una col·lecció de valors (un únic valor per a cada casella buida) la pugui completar convertint-la en SUDOKU-SOLUCIÓ.

I, com que fins ara no hem estat massa explícits (només havíem parlat de triar els sudokus com a tema de recerca i de reorientar l'afició als sudokus des del vessant del consum al de la producció), ja és hora d'aclarir que el propòsit del treball és aprofitar la plataforma de programació en LISP d'AutoCAD (AutoLISP-VisualLISP) per dissenyar un conjunt de funcions que permeti de crear SUDOKUs-PROBLEMA, no pas resoldre'ls (completar-los fins a obtenir les corresponents SUDOKUs-SOLUCIÓ), cosa que en el capítol 12 veurem que SUDÒKULUM (que així es diu el programa, per posar-li un punt d'èmfasi) també està en disposició de fer, però que a l'autor li sembla (tret que només ho usem per verificar la solució) un exercici per a masoquistes: que passi el cafè descafeïnat, la cervesa sense alcohol o la melmelada sense sucre afegit, però furtar-li al joc la seva capacitat d'entretenir és una poca-soltada.

La idea que l'autor va tenir fa quatre anys i mig semblava brillant i simple, tot i que, al cap de poc temps de posar-s'hi, va anar ensumant, cada cop més clar però sense voler-ho reconèixer obertament, que pecava d'irrealitzable:

- Com a tasca prèvia, calia fer allò que anuncia el títol del present capítol: fer l'inventari de totes les SUDOKUs-SOLUCIÓ.
- Disposant ja d'aquest inventari, escolliríem alguna SUDOKU-SOLUCIÓ i passaríem a compondre un dels SUDOKUs-PROBLEMA possibles, mitjançant el mètode següent:
  - Partiríem d'un l'escaquer 9×9 completament buit, situació en relació a la qual totes les SUDOKUs-SOLUCIÓ són compatibles.
  - Aniríem activant caselles (amb això volem dir que les emplenaríem amb el valor predeterminat per la SUDOKU-SOLUCIÓ) i, després de cada activació, examinaríem quantes de les SUDOKUs-SOLUCIÓ compatibles amb la situació precedent seguien sent-ho amb l'actual.

- Quan, després d'una activació, resultés que només quedava una SUDOKU-SOLUCIÓ compatible (que, òbviament, havia de ser l'escollida), la situació ja seria el que podem anomenar un SUDOKU-PROBLEMA estricte, tot i que podríem activar més caselles a costa d'afegir-hi redundància.

Com veieu, la necessitat de disposar de totes les solucions no venia donada tant pel caprici infantil de poder escollir la nostra entre totes les possibles (potser disposant d'un catàleg d'un milió de solucions ja n'hi hagués hagut prou a efectes pràctics) sinó per poder anar comptabilitzant les compatibles amb els emplenaments successius, fins arribar a tenir-ne una de sola: si només en tinguéssim una part, en arribar a aquest hipotètic final sempre ens quedaria el dubte de si, a més de la solució escollida, que finalment havia resultat l'única compatible (única entre les catalogades), no n'hi hauria alguna altra entre les que no havíem inventariat.

I un aclariment: quan dèiem que "escolliríem alguna SUDOKU-SOLUCIÓ i passaríem a compondre un dels SUDOKUs-PROBLEMA possibles" ens referíem al fet que, depenent de l'ordre en què activem les caselles, arribarem a SUDOKUs-PROBLEMA diferents. Per saber de quin ordre de magnitud parlem, podríem suposar una ocupació mitjana dels SUDOKUs-PROBLEMA de 30 caselles: llavors, el nombre s'acostaria al de combinacions **30**-àries de **81** elements (és a dir, a $\mathbf{(81 \times 80 \times ... \times 53 \times 52) / 30!}$). Aproximació dins d'una aproximació, perquè si considerem un conjunt de **N** caselles (**N** pròxim a **30**) i ens limitem a activar les d'aquest conjunt, l'ordre d'activació pot donar lloc a sudokus diferents: per exemple, si activant 30 caselles s'obté un SUDOKU-PROBLEMA estricte, activant-ne una més el SUDOKU-PROBLEMA obtingut serà redundant; però si repetim l'operació mantenint l'ordre d'activació només en les 29 primeres caselles i permutant les dues últimes, es pot produir qualsevol d'aquestes dues situacions:
- Que en resulti un SUDOKU-PROBLEMA estricte de 30 caselles i que l'última casella sigui redundant, com abans.
- Que en resulti un SUDOKU-PROBLEMA estricte de 31 caselles.

Si, en comptes de permutar l'últim emplenament amb el penúltim, el permutem amb l'antepenúltim, llavors pot passar qualsevol d'aquestes tres coses;
- Que en resulti un SUDOKU-PROBLEMA estricte de 29 caselles i que les dues últimes caselles siguin redundants.
- Que en resulti un SUDOKU-PROBLEMA estricte de 30 caselles i que l'última casella sigui redundant, com abans.
- Que en resulti un SUDOKU-PROBLEMA estricte de 31 caselles.

Així doncs, quan més cap enrere (abans) desplacem la casella inicialment redundant (l'activada un cop assolit el SUDOKU-PROBLEMA estricte), més incertesa hi haurà. És per coses com aquesta que no parlarem sovint de SUDOKUs-PROBLEMA estrictes, si no és referint-ho a un ordre d'emplenament concret: la mateixa ocupació que, en activar caselles en un ordre determinat, condueix a un SUDOKU-PROBLEMA estricte, realitzada en un ordre diferent pot dur a un SUDOKU-PROBLEMA redundant (és a dir, a que el SUDOKU-PROBLEMA estricte s'esdevingui abans de completar-la). Fins i tot, en un cas extrem com el que exhibim en l'últim capítol (el lector ja sap que això que llegeix ara ha estat escrit amb l'*Epílog* començat, si més no) ens hem resistit a parlar de "sudoku estricte" per referir-nos a un SUDOKU-PROBLEMA amb 64 caselles ocupades, i hem preferit usar la perífrasi "SUDOKU-PROBLEMA més poblat assolible amb el mètode 2, sense afegir redundància després de convertir-se en única solució compatible".

Aclarit això, reprendrem l'estratègia de què ens ocupem en el present capítol i de la qual just havíem iniciat la presentació, per aportar una mica de serenitat a la primera reacció del lector, que sens dubte haurà estat d'hilaritat o d'indignació, segons el seu tarannà, en veure tanta desmesura. I ho farem dient que l'estratègia admet un parell de modificacions que simplifiquen notablement el procés... però no tant com per transformar-lo d'utòpic en realitzable:
- Tret que es vulgui orientar l'usuari, fent que estigui constantment informat del resultat de cada activació, no cal estendre la comparació a totes les solucions: trobant-ne alguna altra de compatible, a més de l'original (la SUDOKU-SOLUCIÓ), ja podríem continuar, activant més caselles fins a quedar-nos només amb aquesta.
- Si anomenem SUDOKUs-SOLUCIÓ normalitzades aquelles en què es compleixen les dues condicions següents,
  - estar ordenada cadascuna de les tres tríades, per files,
  - estar ordenat el conjunt de les tres tríades, atenent a les primeres files,

  no cal disposar de totes les SUDOKUs-SOLUCIÓ, sinó només de les normalitzades: com que qualsevol solució es pot normalitzar reordenant les files de cada tríada i tot seguit reordenant les tríades, quan haguem obtingut un SUDOKU-PROBLEMA de la solució normalitzada, podrem restituir-la aplicant les permutacions inverses.

Incorporant aquestes simplificacions, descriurem així el procediment:
- Com a tasca prèvia, farem inventari de totes les SUDOKUs-SOLUCIÓ normalitzades.
- Disposant ja d'aquest inventari, n'escollirem alguna SUDOKU-SOLUCIÓ i passarem a compondre un dels SUDOKUs-PROBLEMA possibles, mitjançant el mètode següent:
  - Partirem d'un escaquer 9×9 completament buit, situació en relació a la qual totes les SUDOKUs-SOLUCIÓ normalitzades són compatibles.
  - Anirem activant caselles i, després de cada activació, explorarem el catàleg de solucions normalitzades fins a trobar-ne alguna, diferent de l'escollida, que sigui compatibles amb la situació actual.
  - Quan, després d'una activació, ja no trobem cap solució nomalitzada, diferent de l'escollida, que sigui compatible amb el conjunt de les caselles activades, tindrem un SUDOKU-PROBLEMA.

Però pensant-hi una mica, us adonareu que les millores són més aparents que reals: d'una banda, treballar amb les solucions normalitzades representa fer-ho amb la 1.296-èsima part del conjunt de SUDOKUs-SOLUCIÓ (tant el nombre de permutacions de files en cada tríada com el de tríades en la solució és de **3!= 6**, o sigui que el nombre de solucions que, en ser normalitzades, acabaran resultant idèntiques, serà de $\mathbf{6^4 = 1.296}$), cosa que en termes relatius pot semblar un avenç considerable però que, en termes absoluts, segueix sent inabastable; de l'altra, a la llarga ningú no ens lliurarà de tenir-nos-les amb totes les solucions (normalitzades i sense normalitzar, com veurem en el capítol 11), però encara és més important adonar-se que en un moment o altre hauríem d'examinar la compatibilitat de la situació amb totes les solucions (en començar, no costarà gens de trobar una solució compatible i diferent de l'adoptada, peró cada cop costarà mes, i al final no hi hauria altre remei que examinar-les totes per assegurar que no n'hi ha cap més, de compatible). Les alternatives les discutirem entre els capítols 8 i 12, però podem avançar que passen per canviar de xip buscant les solucions compatibles durant el procés, no pas explorant un enorme catàleg previ per buscar-hi les compatibilitats, i llavors cada cop costarà menys de trobar una solució compatible, en comptes de costar més (proseguint en la línia de buscar-ne alguna), o bé per començar el recompte amb un mínim nombre de caselles activades, no pas amb l'escaquer 9×9 buit (proseguint en la línia de saber quantes solucions compatibles queden després de cada activació).

Però som aquí per veure com ens ho vam muntar fa quatre anys i mig, fins on es va arribar i com *a posteriori* s'ha pogut millorar part de la feina feta (si no per al propòsit inicial, sí que l'hem deixada en condicions de prestar algun servei), de manera que seguirem l'exposició com si el camí emprès hagués de dur a algun lloc. Així que desconnectarem de la qüestió de què en faríem, si efectivament haguéssim pogut reunir l'inventari monstruós (eludint també el problema de quina estructura d'indexació donaríem als arxius que suportessin aquesta informació), centrant-nos en la seva construcció. Per suposat, ens acollirem als petits avantatges de jugar amb SUDOKUs-SOLUCIÓ normalitzades, que no són únicament el seu menor nombre sinó sobretot el menor nombre de tríades normalitzades inferiors, centrals i superiors. Tenint en compte que, tant si considerem les files valors numèrics (**123.56.789** a **987.654.321**) com textos (**"1234567890"** a **"987654321"**), en la ordenació a efectes de complir els requisits establerts perquè una SUDOKU-SOLUCIÓ sigui normalitzada ens podem limitar a la primera xifra o caràcter, atès que totes consten de 9 elements, les condicions que hauran de complir els requadres 9×3 són, optant per les xifres:
- Primera tríada:
  - La 1ª fila ha de començar per **1**, per definició.
  - La 2ª fila pot començar per qualsevol valor entre **2** i **8** (si comencés per **9**, no quedaria cap valor per a la 3ª).
  - La 3ª fila pot començar per qualsevol valor entre **3** i **9**.
- Segona tríada:
  - La 4ª fila pot començar per qualsevol valor entre **2** i **4** (si comencés per **5**, la 5ª i 6ª fila haurien de començar per valors superiors, i també la 7ª, 8ª i 9ª fila, cosa que no pot ser sense repetir valors, perquè només en queden quatre: **6**, **7**, **8** i **9**).
  - La 5ª fila pot començar per qualsevol valor entre **3** i **8** (si comencés per **9**, no quedaria cap valor per a la 6ª).
  - La 6ª fila pot començar per qualsevol valor entre **4** i **9**.
- Tercera tríada:
  - La 7ª fila pot començar per qualsevol valor entre **3** i **7** (si comencés per **8**, la 8ª i 9ª fila haurien de començar per valors superiors, i només queda el **9**).
  - La 8ª fila pot començar per qualsevol valor entre **4** i **8** (si comencés per **9**, no quedaria cap valor per a la 9ª).
  - La 9ª fila pot començar per qualsevol valor entre **5** i **9**.

En resum:
- La 1ª fila ha de començar amb **1**.
- La 2ª fila pot començar amb valors entre **2** i **8**.
- La 3ª fila pot començar amb valors entre **3** i **9**.
- La 4ª fila pot començar amb valors entre **2** i **4**.
- La 5ª fila pot començar amb valors entre **3** i **8**.
- La 6ª fila pot començar amb valors entre **4** i **9**.
- La 7ª fila pot començar amb valors entre **3** i **7**.
- La 8ª fila pot començar amb valors entre **4** i **8**.
- La 9ª fila pot començar amb valors entre **5** i **9**.

El primer pas va ser la construcció d'un arxiu ASCII amb les **9!= 362.880** files, ocupant cada una un registre. Com que aquest arxiu s'anomenaria 123456789.txt, la funció creadora, caracteritzada com a ordre d'AutoCAD, es va dir **C:FES-123456789**. Semblava obligat muntar-la per formar totes les permutacions dels elements de la llista **("1" "2" "3" "4" "5" "6" "7" "8" "9")** i guardar-les encadenades en textos, però vam veure que resultava més senzill partir del valor numèric **123.456.789**, que incrementaríem d'un en un (864.197.532 vegades) fins arribar al valor **987.654.321**: dels enters que anessin sortint n'exclouríem, previament convertits en textos, els que continguessin algun **"0"** o qualsevol altre caràcter numèric repetit; la segona exclusió, al seu torn, resultava més fàcil d'escometre copsant l'absència d'algun dels 9 caràcters (en un conjunt de nou caràcters **"1"**... **"9"** tots son diferents o, si n'hi ha algun de repetit, n'ha de faltar un altre). Aquí teniu aquesta funció:

```
(defun C:FES-123456789 (/ A M N)
  (setq A (open "123456789.txt" "w")
        M 123456788)
  (repeat 864197533
    (if (wcmatch (setq M (1+ M) N (itoa M)) "*0*")
        (setq N ())
        (foreach C '("1" "2" "3" "4" "5" "6" "7" "8" "9")
          (if (and N (not (wcmatch N (strcat "*" C "*")))) (setq N ()))))
    (if N (write-line N A)))
  (close A))
```

L'arxiu 123456789.txt ocupa **3.899 KBytes** i va trigar gairebé **2 hores** a formar-se. Val a dir que aquest temps no és massa representatiu perquè correspon a un *Pentium* adquirit cap a l'any 2000 i apedaçat diverses vegades (quan escriu aquestes línies l'autor disposa d'un portàtil del 2010, en el qual el procés ha trigat **1 hora i 19 minuts**, però d'aquesta història del nou equip és millor que no en tornem a parlar fins el capítol 13) i tampoc no ens ha de preocupar la durada excessiva, perquè es tracta d'un arxiu singular. Més preocupant va ser la durada i l'ocupació en disc dels arxius de tríades normalitzades, però abans de passar a aquest tema acabarem amb l'arxiu de línies. Com és lògic, necessitàvem llegir l'arxiu 123456789.txt per transferir-ne el contingut a una llista **L1** que poguéssim manipular, i vam crear la funció **C:CARREGA-123456789** amb aquest propòsit:

```
(defun C:CARREGA-123456789 (/ A)
  (setq RUTA (findfile "123456789.txt") A (open RUTA "r") L1 ())
  (repeat 362880 (setq L1 (cons (read-line A) L1)))
  (setq L1 (reverse L1))
  (close A))
```

Observeu que, ni la llista **L1** ni l'adreça **RUTA** (si 123456789.txt no se situa a la ruta de cerca d'AutoCAD, **C:CARREGA-123456789** donarà error) no estan declarades en la definició, perquè bàsicament la funció serà accedida des d'altres funcions on sí que declararem aquestes variables. Si, malgrat això, **C:CARREGA-123456789** també està caracteritzada com a ordre d'AutoCAD, és per fer-li més còmoda la utilització a l'usuari, que sovint haurà de carregar 123456789.txt perquè voldrà saber quina fila correspon a una determinada posició **N**, fent **(nth N L1)** o, més probablement, resoldre el problema invers: conèixer quina posició ocupa a la llista **L1** una fila determinada, mitjançant les funcions auxiliars **LOCL1** i **LOC1A9L1**. En concret, fent **(LOCL1** <contingut de l'element N-èsim de L1>**)** obtenim la posició **N** d'una fila a **L1** (la línia cal subministrar-la en forma de text, i el primer element correspon a la posició **N** = **0**); així que, si **F** és el contingut de la fila, **(nth (LOCL1 F) L1)** = **F**. **LOC1A9L1** crea la llista **LOC**, directori de les posicions a **L1** on comencen les files que tenen "1", "2", ... "9" com a primer caràcter; estrictament, la llista **LOC** és **(0 40320 80640 120960 161280 201600 241920 282240 322560)**, i té la particularitat

que entre elements consecutius l'intèrval és constant i val **362.880 / 9 = 40.320**.
Les files inicials i finals de cada intèrval (amb les seves posicions a **L1**) són:

"**1**23456789" a la posició 0 = **(nth 0 LOC)** i "**1**98765432" a la 40.319 (40.320 files)
"**2**13456789" " 40.320 = **(nth 1 LOC)** i "**2**98765431" a la 80.639 (40.320 ")
"**3**12456789" " 80.640 = **(nth 2 LOC)** i "**3**98765421" a la 120.959 (40.320 ")
"**4**12356789" " 120.960 = **(nth 3 LOC)** i "**4**98765321" a la 161.279 (40.320 ")
"**5**12346789" " 161.280 = **(nth 4 LOC)** i "**5**98764321" a la 201.599 (40.320 ")
"**6**12345789" " 201.600 = **(nth 5 LOC)** i "**6**98754321" a la 241.919 (40.320 ")
"**7**12345689" " 241.920 = **(nth 6 LOC)** i "**7**98654321" a la 282.239 (40.320 ")
"**8**12345679" " 282.240 = **(nth 7 LOC)** i "**8**97654321" a la 322.559 (40.320 ")
"**9**12345678" " 322.560 = **(nth 8 LOC)** i "**9**87654321" a la 362.879 (40.320 ")
(TOTAL = 40.320 × 9 = 362.880 files)

Per raons pràctiques (no haver d'estar pendents del perill que algun bucle dugués a l'execució de **(nth LOC 9)** i això fos causa d'error en funcions amb accés a **LOC**), en aquesta llista li vam afegir el desè element **362.880**. Veieu les dues funcions:

```
(defun LOCL1 (E) (- 362880 (length (member E L1))))

(defun LOC1A9L1 (/ PRIMS)
  (setq PRIMS '("213456789" "312456789" "412356789" "512346789"
                "612345789" "712345689" "812345679" "912345678")
        LOC '(0))
  (foreach P PRIMS (setq LOC (cons (LOCL1 P) LOC)))
  (setq LOC (cons 362880 LOC) LOC (reverse LOC)))
```

Ja teníem la informació de base (tot allò relatiu a les files), però uns objectius tan ambiciosos com enregistrar totes les SUDOKUs-SOLUCIÓ de segur que requeririen elaborar aquesta informació per disposar de dades intermèdies i evitar, dintre del que fos raonable i possible, la repetició de processos idèntics. I la primera fase d'aquesta preparació artillera era sens dubte determinar la compatibilitat entre files i guardar-la en un arxiu. Ho confiaríem a una funció tan simple com aquesta:

```
(defun C:FES-M-COMPAT-N (/ A K M N O FM FN COMP L1 LOC CC)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq A (open "M-COMPAT-N.txt" "w") M -1 K 1)
  (while (< (setq M (1+ M)) 322560)        ; 322560 = (nth 8 LOC)
         (if (= M (nth K LOC)) (setq K (1+ K)))
         (setq FM (nth M L1) N (1- (nth K LOC)) CC ())
         (while (< (setq N (1+ N)) 362880) ; 362880 = (nth 9 LOC)
                (setq FN (nth N L1) O 1 COMP T)
                (while (and COMP (< O 9))
                       (setq O (1+ O)
                             COMP (/= (substr FM O 1) (substr FN O 1))))
                (if COMP (setq CC (strcat (if CC CC (itoa M)) " " (itoa N)))))
         (if CC (write-line CC A)))
  (close A))
```

Amb **C:FES-M-COMPAT-N** volíem crear un arxiu M-COMPAT-N.txt que, per a cada fila de **L1** (representada per la seva posició **M**, per estalviar espai), indiqués les files compatibles, en el sentit de no haver-hi coincidències en columna (representades també per la posició **N** a **L1**). A cada registre hi hauria en primer lloc la fila de referència (0, 1, 2... 322.559), seguida per les files compatibles (algunes podien repetir-se en altres registres). La finalitat era avançar feina, tant a l'hora de fer inventari de les tríades de files compatibles (evitant cercar directament a **L1** i reduint així el nombre de files que calgués ignorar en trobar-s'hi coincidències per tercets) com a l'hora d'associar a cadascuna d'aquestes tríades una llista de files compatibles per columnes (les presents a tots tres registres, tret d'elles mateixes), per tenir-ho més fàcil a l'hora final d'inventariar les SUDOKUs-SOLUCIÓ (detectar la compatibilitat de tres tríades de files compatibles esdevé una feina més planera quan es limita a verificar si les files que formen la 2ª figuren en la llista associada a la 1ª, i les que formen la 3ª en les associades a la 1ª i 2ª). Però el nombre de files compatibles amb les primeres files de **L1** era elevadíssim (cada línia de les set primeres trigava **25 minuts** i ocupava gairebé un MegaByte) i, tot i que a cada nova fila s'hagués anat reduint (amb solucions normalitzades, les files només es comparen amb les situades més endavant), el procés hauria durat anys i M-COMPAT-N.txt hauria tingut una mida brutal, raó per la qual vam desistir.

En vista d'això, i en comptes de donar per exhaurida aquesta via, vam pensar que l'obstacle del creixement sense mesura de M-COMPAT-N.txt podia pal·liar-se amb un grau d'exigència més alt, que reduís sensiblement el nombre de files compatibles amb cada fila: en comptes de prendre l'accepció de compatibilitat referida a la no coincidències en columna (aplicable entre les files situades en diferents tríades d'una SUDOKU-SOLUCIÓ), prendríem la relativa a no coincidències en un requadre 3×3 (aplicable entre les files situades en una mateixa tríada), canvi que afavoriria, a més, l'inventari final de solucions (tríades compatibles de files compatibles). Però no sabíem si la major laboriositat del nou procediment es compensaria amb el menor nombre de files compatibles trobades (registres del nou arxiu M-COMPAT-N.txt més curts), perquè al capdavall el nombre de comparacions era el mateix, tot i que no costava massa descartar, d'entrada, no només les files **N** amb el primer caràcter inferior o igual al primer de la fila **M**, sinó igual als caràcters segon o tercer, dispositiu mitjançant el qual passavem d'un nombre de comparacions en profunditat $8\times 40.320^2 + 7\times 40.320^2 + 6\times 40.320^2 + 5\times 40.320^2 + 4\times 40.320^2 + 3\times 40.320^2 + 2\times 40.320^2 + 40.320^2 = (8+7+6+5+4+3+2+1)\times 40.320^2 = 36\times 40.320^2 = 58.525.286.400$ a "només" $0{,}75\times(8+7+6+5+4+3+2+1)\times 40.320^2 = 27\times 40.320^2 = 43.893.964.800$. Convé fer un incís perquè no sembli que ens hem tret aquests valors de la màniga:

- Quan d'entrada apartem únicament les files **FN** que comencen pel mateix caràcter numèric I_1 que **FM**, aquesta només l'hauríem de comparar amb les $(9-I_1)\times 40.320$ files que comencen amb un caràcter $J > I_1$.
- Si també volem apartar les **FN** que comencen amb el segon o tercer caràcter de **FM**, I_2 i I_3, únicament quan sigui $I_1 < I_2$ i $I_1 < I_3$ podrem limitar-nos a comparar **FM** amb les $(9-I_1-2)\times 40.320$ files **FN** que comencin amb caràcters superiors a I_1, I_2 i I_3; però també pot ser que $I_2 < I_1 < I_3$ o $I_2 < I_1 < I_3$, cas en què el nombre de files **FN** a comparar amb **FM** serà $(9-I_1-1)\times 40.320$, o que $I_2 < I_1$ i $I_3 < I_1$, cas en què el nombre de files **FN** a comparar amb **FM** serà $(9-I_1)\times 40.320$. Així doncs, cadascuna de les **40.320** files que comencen amb I_1 haurà de ser comparada amb un nombre diferent de files **FN**, però sí que podem calcular, per a cada I_1, quantes ho fan amb $(9-I_1-2)\times 40.320$ files **FN**, quantes amb $(9-I_1-1)\times 40.320$ i quantes amb $(9-I_1)\times 40.320$, perquè, en el conjunt de les **40.320** files, els 8 caràcters diferents de I_1 es reparteixen equitativament el segon i tercer lloc: només cal determinar la freqüència d'aquestes tres situacions, a partir de la freqüència $(9-I_1)\times 40.320$ de les files situades per sobre del conjunt I_1 i de la freqüència $(I_1-1)\times 40.320$ de les situades per sota. Aquí en teniu la taula de valors (s'ha omès el factor **40.320** per poder-la encabir en l'amplada de pàgina):

	$I_1 < I_2$ i $I_1 < I_3$	$I_2 < I_1 < I_3$ o $I_2 < I_1 < I_3$	$I_2 < I_1$ i $I_3 < I_1$
$I_1 = 9$:	$\frac{0}{8} = 0$	$2\times\frac{8}{8}\times\frac{0}{7} = \frac{0}{56} = 0$	$\frac{8}{8}\times\frac{7}{7} = \frac{56}{56} = 1$
$I_1 = 8$:	$\frac{1}{8}\times\frac{0}{7} = \frac{0}{56} = 0$	$2\times\frac{7}{8}\times\frac{1}{7} = \frac{14}{56} = 0{,}25$	$\frac{7}{8}\times\frac{6}{7} = \frac{42}{56} = 0{,}75$
$I_1 = 7$:	$\frac{2}{8}\times\frac{1}{7} = \frac{2}{56} = 0{,}0357$	$2\times\frac{6}{8}\times\frac{2}{7} = \frac{24}{56} = 0{,}4286$	$\frac{6}{8}\times\frac{5}{7} = \frac{30}{56} = 0{,}5357$
$I_1 = 6$:	$\frac{3}{8}\times\frac{2}{7} = \frac{6}{56} = 0{,}1071$	$2\times\frac{5}{8}\times\frac{3}{7} = \frac{30}{56} = 0{,}5357$	$\frac{5}{8}\times\frac{4}{7} = \frac{20}{56} = 0{,}3571$
$I_1 = 5$:	$\frac{4}{8}\times\frac{3}{7} = \frac{12}{56} = 0{,}2143$	$2\times\frac{4}{8}\times\frac{4}{7} = \frac{32}{56} = 0{,}5714$	$\frac{4}{8}\times\frac{3}{7} = \frac{12}{56} = 0{,}2143$
$I_1 = 4$:	$\frac{5}{8}\times\frac{4}{7} = \frac{20}{56} = 0{,}3571$	$2\times\frac{3}{8}\times\frac{5}{7} = \frac{30}{56} = 0{,}5357$	$\frac{3}{8}\times\frac{2}{7} = \frac{6}{56} = 0{,}1071$
$I_1 = 3$:	$\frac{6}{8}\times\frac{5}{7} = \frac{30}{56} = 0{,}5357$	$2\times\frac{2}{8}\times\frac{6}{7} = \frac{24}{56} = 0{,}4286$	$\frac{2}{8}\times\frac{1}{7} = \frac{2}{56} = 0{,}0357$
$I_1 = 2$:	$\frac{7}{8}\times\frac{6}{7} = \frac{42}{56} = 0{,}75$	$2\times\frac{1}{8}\times\frac{7}{7} = \frac{14}{56} = 0{,}25$	$\frac{1}{8}\times\frac{0}{7} = \frac{0}{56} = 0$
$I_1 = 1$:	$\frac{8}{8}\times\frac{7}{7} = \frac{56}{56} = 1$	$2\times\frac{0}{8} = 0$	$\frac{0}{8} = 0$

Pel que fa al nombre de files **FN** amb què es comparen les **FM**, segons I_1 i les posicions relatives de I_2 i I_3 (amb les excepcions oportunes a les expressions genèriques $(9-I_1-2)\times 40.320$, $(9-I_1-1)\times 40.320$ i $(9-I_1)\times 40.320$), veieu-les:

	$(9 - I_1 - 2) \times 40.320$	$(9 - I_1 - 1) \times 40.320$	$(9 - I_1) \times 40.320$
$I_1 = 9$:	0×40.320	0×40.320	0×40.320
$I_1 = 8$:	0×40.320	0×40.320	1×40.320
$I_1 = 7$:	0×40.320	1×40.320	2×40.320
$I_1 = 6$:	1×40.320	2×40.320	3×40.320
$I_1 = 5$:	2×40.320	3×40.320	4×40.320
$I_1 = 4$:	3×40.320	4×40.320	5×40.320
$I_1 = 3$:	4×40.320	5×40.320	6×40.320
$I_1 = 2$:	5×40.320	6×40.320	0×40.320
$I_1 = 1$:	6×40.320	0×40.320	0×40.320

Si ara multipliquem els ítems homòlegs (recuperant el factor **40.320** omès en la primera taula) i, per a cada I_1, sumem els termes corresponents a cadascuna de les tres posicions relatives de I_2 i I_3, obtindrem aquests valors,

$I_1 = 9$: $(0 + 0 + 0) \times 40.320^2 = 0 \times 40.320^2$

$I_1 = 8$: $(0 + 0 + 0{,}75 \times 1) \times 40.320^2 = 0{,}75 \times 40.320^2$

$I_1 = 7$: $(0 + 0{,}4286 \times 1 + 0{,}5357 \times 2) \times 40.320^2 = 1{,}5 \times 40.320^2$

$I_1 = 6$: $(0{,}1071 \times 1 + 0{,}5357 \times 2 + 0{,}3571 \times 3) \times 40.320^2 = 2{,}25 \times 40.320^2$

$I_1 = 5$: $(0{,}2143 \times 2 + 0{,}5714 \times 3 + 0{,}2143 \times 4) \times 40.320^2 = 3 \times 40.320^2$

$I_1 = 4$: $(0{,}3571 \times 3 + 0{,}5357 \times 4 + 0{,}1071 \times 5) \times 40.320^2 = 3{,}75 \times 40.320^2$

$I_1 = 3$: $(0{,}5357 \times 4 + 0{,}4286 \times 5 + 0{,}0357 \times 6) \times 40.320^2 = 4{,}5 \times 40.320^2$

$I_1 = 2$: $(0{,}75 \times 5 + 0{,}25 \times 6 + 0) \times 40.320^2 = 5{,}25 \times 40.320^2$

$I_1 = 1$: $(1 \times 6 + 0 + 0) \times 40.320^2 = 6 \times 40.320^2$

generalitzables mitjançant l'expressió $0{,}75 \times (9 - I_1) \times 40.320^2$. Per això la suma és $0{,}75 \times (8 + 7 + 6 + 5 + 4 + 3 + 2 + 1) \times 40.320^2 = 27 \times 40.320^2 = 43.893.964.800$.

Donem per tancat l'incís recordant allò que havíem dit en relació a molts estalvis d'operacions arxidemostrats i, malgrat tot, no reflectits en la durada del procés. Així que, per sortir de dubtes, vam dissenyar una nova versió de **C:FES-M-COMPAT-N**, que per la seva major complexitat incorporava les funcions auxiliars **3*3** i **COMPAT**:

```
(defun 3*3 (J / LM LN)
  (repeat 3 (setq J (1+ J) LM (cons (nth J LTM) LM) LN (cons (nth J LTN) LN)))
  (not (or (member (car LN) LM) (member (cadr LN) LM) (member (last LN) LM))))

(defun COMPAT (/ LTN COMP TMJ TNJ)
  (setq LTN (list (substr FN 1 1)) COMP T J 1)
  (while (and COMP (< J 9))
         (setq TMJ (nth J RLTM) J (1+ J) TNJ (substr FN J 1)
               COMP (/= TMJ TNJ))
         (if COMP (setq LTN (cons TNJ LTN))))
  (and COMP (3*3 -1) (3*3 2) (3*3 5)))

(defun C:FES-M-COMPAT-N (/ A I I1 I2 I3 J K M N FM FN LTM RLTM L1 LOC CC)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq A (open "M-COMPAT-N.txt" "w") M -1 K 1)
  (while (< (setq M (1+ M)) 322560) ; 322560 = (nth 8 LOC)
         (setq FM (nth M L1)
               I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
               LTM (list I3 I2 I1) J 3)
         (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
         (setq RLTM (reverse LTM))
         (if (> I1 I2) (setq I I1 I1 I2 I2 I))
         (if (> I2 I3)
             (progn
               (setq I I2 I2 I3 I3 I)
               (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
         (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
               K (car I) CC ())
```

```
        (while (< (setq K (1+ K)) 9)
               (foreach J I (if (= K J) (setq K (1+ K))))
               (if (< K 9)
                   (progn
                     (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC))
                     (while (< (setq N (1+ N)) I1)
                            (setq FN (nth N L1))
                            (if (COMPAT) (setq CC (strcat (if CC CC (itoa M))
                                                          " " (itoa N))))))))
        (if CC (write-line CC A)))
  (close A))
```

El resultat no va ser decebedor del tot: referint-nos, com abans, a cada registre dels set primers, havíem passat de **25** a **10 minuts** i d'una ocupació de 890 KBytes a 83 KBytes, de mitjana. Tanmateix el temps seguia sent excessiu perquè el projecte fos viable i, en un últim intent de descobrir alguna llei oculta que permetés fer prediccions sobre la compatibilitat d'algun parell de files (ja no parlem de tots) sense haver de passar per l'adreçador **COMPAT**, vam trobar dues formes d'aparellar les 362.880 files (podríem parlar d'aplicacions bijectives, però l'autor s'estima més no entrar en un terreny que li queda molt llunyà) que servien aquest propòsit:

- Anomenant "files simètriques" aquelles en què cada caràcter de la primera ocupa la posició simètrica en la segona, definíem la correspondencia entre continguts simètrics, **FM** i **FN**, mitjançant la funció reversible **SIMETR** (**(SIMETR FM)** = **FN** i **(SIMETR FN)** = **FM**), i la correspondència entre les posicions d'aquestes files a **L1**, **M** i **N**, amb la funció **SIM**, igualment reversible (**(SIM M)** = **N** i **(SIM N)** = **M**).
- Anomenant "files complementàries" aquelles en què la suma dels valors numèrics representats pels caràcters homòlegs és 10, definíem la correspondencia entre continguts complementaris, **FM** i **FN**, mitjançant una funció reversible que (per no confondre **COMP10** amb "complement de deu", en l'accepció computacional) va acabar rebent el nom **DIFA10** (**(DIFA10 FM)** = **FN** i **(DIFA10 FN)** = **FM**), i la correspondència entre les posicions d'aquestes files a **L1**, **M** i **N**, amb la funció **DIF**, igualment reversible (**(DIF M)** = **N** i **(FID N)** = **M**).

Observeu que la primera i l'última fila de **L1**, "**123456789**" i "**987654321**", són a la vegada simètriques i complementàries, tot i que aquest no era el motiu (no l'únic, almenys) de les esperances que teníem dipositades en aquestes correspondències als efectes d'estalviar-nos comparacions en profunditat entre files (accés a **COMPAT**). Les propietats que podien ser-nos útils, comuns a **SIMETR** i **DIFA10**, eren aquestes:

- La fila **FM** mai no és compatible amb **(SIMETR FM)** ni amb **(DIFA10 FM)**: en el primer cas, per la coincidència del tres caràcters del tercet central i, en el segon, pels dos caràcters "**5**" sempre situats a la mateixa columna. Exemple: "**257491386**" és incompatible amb "**683194752**" (simètrica) i amb "**853619724**" (complementària).
- Si **FM** i **FN** són compatibles, **(SIMETR FM)** i **(SIMETR FN)** també ho són, i el mateix passa amb **(DIFA10 FM)** i **(DIFA10 FN)**. En el cas de la simetria la justificació és immediata: els caràcters en columna segueixen estant-ho i el tercets segueixen corresponent-se (només han variat nominalment: els primers han passat a ser els tercers, i viceversa), de forma que no poden haver-hi coincidències perquè abans no n'hi havia. I, en el cas de files complementàries, si entre tercets homòlegs s'esdevé que $\mathbf{a} \neq \mathbf{b}$ (per a qualsevol dels 3 caràcters **a** d'una fila i qualsevol dels 3 caràcters **b** de l'altra), ha de ser $\mathbf{-a \neq -b}$ i $\mathbf{10-a \neq 10-b}$. Exemple: com que "**148973256**" i "**692584713**" són compatibles, "**652379841**" i "**317485296**" (simètriques de les primeres) també ho són, i el mateix passa entre "**962137854**" i "**418526397**" (complementàries de les primeres).
- Si **FM** i **FN** no són compatibles, **(SIMETR FM)** i **(SIMETR FN)** tampoc no ho són, i el mateix passa amb **(DIFA10 FM)** i **(DIFA10 FN)**. En el cas de la simetria només cal repetir les consideracions d'abans per concloure que les coincidències entre carácters de tercets homòlegs es mantenen. En el cas de files complementàries, si entre tercets homòlegs s'esdevé que $\mathbf{a = b}$ (per a algun dels 3 caràcters **a** d'una fila i dels 3 caràcters **b** de l'altra) ha de ser $\mathbf{-a = -b}$ i $\mathbf{10-a = 10-b}$. Exemple: com que "**391782456**" i "**529481637**" no són compatibles, "**654287193**" i "**736184925**" (simètriques de les primeres) tampoc no ho són, i el mateix passa entre "**719328654**" i "**581629473**" (complementàries de les primeres).

Recordant que, per definició, **M** < **N** (**FM** està situada a **L1** abans que **FN**) però que també és **FM** < **FN** (raó per la qual ens limitem a parlar de continguts de files, per no complicar més les coses), i adoptant l'estrany nom híbrid **SIMDIF** per no ser tan reiteratius, fent aserts sobre **SIMETR** i repetint-los sobre **DIFA10**, veiem com pot intervenir **SIMDIF** (com poden intervenir **SIMETR** i **DIFA10**) fent innecessaris alguns accessos a l'entretinguda rutina **COMPAT**, durant la construcció de M-COMPAT-N.txt:

- En comparar **FM** amb **FN** = **(SIMDIF FM)** (com que també **FM** = **(SIMDIF FN)**, l'atribució de noms ja s'ha fet de forma que **FM** < **(SIMDIF FM)**) i estar assabentats d'aquesta circumstància, no cal anar a **COMPAT**, perquè sabem que **FM** és incompatible amb **FN**.
- Pel que fa a la segona i tercera propietats, la primera comparació (**FM** amb **FN**) serà la de "sembra" (**COMPAT**arar **FM** i **FN**, guardant el resultat i les posicions de **(SIMDIF FM)** i **(SIMDIF FN)**), lògicament, i la segona (**(SIMDIF FM)** amb **(SIMDIF FN)** o **(SIMDIF FM)** amb **(SIMDIF FN)**, segons que **FM** < **(SIMDIF FM)** < **(SIMDIF FN)** o **FM** < < **(SIMDIF FN)** < **(SIMDIF FM)**) la de "collita" (aplicar a les files corresponents a aquestes posicions el resultat emmagatzemat). L'única dificultat serà activar i desactivar els senyals que ens obligaran a estar pendents del moment en què la primera fila en la comparació sigui **(SIMDIF FM)** (o **(SIMDIF FN)**) i la segona fila sigui **(SIMDIF FN)** (o **(SIMDIF FM)**).

L'avaluació del rendiment obtingut en la construcció de M-COMPAT-N.txt gràcies a la implementació d'aquests recursos l'havíem de fer experimentalment (ni que fos comparant durades en una minúscula etapa del procés; limitació obligada, d'altra banda), perquè sovint és pitjor el remei que la malaltia. Vam provar per separat la intervenció de **SIMETR** i de **DIFA10**, començant per definir la primera funció:

```
(defun SIMETR (TXT1 / K TXT2)
  (setq TXT2 "" K 10)
  (repeat 9
    (setq K (1- K)
          TXT2 (strcat TXT2 (substr TXT1 K 1)))))
```

Fet això, vam posar-nos a obtenir parells de valors i, contra les expectatives que obria la constatació que la primera i l'última fila de **L1** eren simètriques, no vam saber descobrir cap llei que lligués les posicions de les files simètriques i que pogués plasmar-se en una funció **SIM** solvent, més enllà de la carrinclona i lenta definició literal **(defun SIM (M) (LOCL1 (SIMETR (nth M L1))))**. Per això, com que una versió com aquesta no era viable si havia de ser executada tan reiteradament com esperàvem fer-ho a la funció **C:FES-M-COMPAT-N**, més valia que ens molestéssim una vegada per sempre a construir un arxiu SIMETR.TXT amb l'enumeració exhaustiva de les correspondències i que, descarregat en una llista **LS**, permetés d'obtenir la posició **SIMM** de la fila simètrica de **(nth M L1)**, fent només **(setq SIMM (nth M LS))**. Per conjurar possibles imputacions de poca perspicàcia, el millor serà oferir una mostra d'aquest arxiu, amb les 20 primeres posicions **SIMM**, les 20 del mig i les 20 finals (precedides de les posicions **M**, entre parèntesis), abans de presentar les funcions **C:FES-SIMETR** i **CARREGA-SIMETR** (que creen SIMETR.txt i en transfereixen el contingut a **LS**) i, si després d'examinar-lo, encara hi ha algun lector que afirmi haver descobert una llei, l'exhortem a que la presenti en algun congrés matemàtic:

```
                        ......... .......           ......... .......
      (0) 362.879       (181.430) 201.664           (362.860)  86.400
      (1) 322.559       (181.431)  20.224           (362.861)  46.080
      (2) 357.839       (181.432)  42.544           (362.862) 131.040
      (3) 282.239       (181.433)   2.224           (362.863)  90.720
      (4) 317.519       (181.434)  85.984           (362.864) 121.680
      (5) 277.199       (181.435)  45.664           (362.865)  10.800
      (6) 362.159       (181.436)  80.944           (362.866)  81.360
      (7) 321.839       (181.437)   5.344           (362.867)   5.760
      (8) 352.799       (181.438)  40.624           (362.868) 126.000
      (9) 241.919       (181.439)     304           (362.869)  50.400
     (10) 312.479       (181.440) 362.575           (362.870) 120.960
     (11) 236.879       (181.441) 322.255           (362.871)  10.080
     (12) 357.119       (181.442) 357.535           (362.872)  41.040
     (13) 281.519       (181.443) 281.935           (362.873)     720
     (14) 352.079       (181.444) 317.215           (362.874)  85.680
     (15) 241.199       (181.445) 276.895           (362.875)  45.360
     (16) 272.159       (181.446) 360.655           (362.876)  80.640
     (17) 231.839       (181.447) 320.335           (362.877)   5.040
     (18) 316.799       (181.448) 342.655           (362.878)  40.320
     (19) 276.479       (181.449) 161.215           (362.879)       0
     .... .......       ......... .......
```

```
(defun C:FES-SIMETR (/ A M)
  (C:CARREGA-123456789)
  (setq A (open "SIMETR.txt" "w") M -1)
  (repeat 362880
    (setq M (1+ M))
```

```
    (write-line (itoa (LOCL1 (SIMETR (nth M L1)))) A))
  (close A))

(defun CARREGA-SIMETR ()
  (setq A (open "SIMETR.txt" "r"))
  (repeat 362880 (setq LS (cons (atoi (read-line A)) LS)))
  (setq LS (reverse LS))
  (close A))
```

Passant a **C:FES-M-COMPAT-N**, la primera meitat de la funció només es diferenciava de la versió precedent per la descàrrega de SIMETR.txt a la llista **LS**, mitjançant la **CARREGA-SIMETR** que acabem de presentar, l'ús d'aquesta llista en l'assignació dels valors **(nth M LS)** i **(nth M LS)** a **SIMM** i **SIMN**, respectivament (aquesta última ja en la segona meitat), i en un altra assignació que convé explicar: la del valor binari **(< M SIMM)** a la variable **PEND** ja que, en cada iteració **M**, **PEND** indicava si havíem d'estar pendents del moment en què **N** coincidís amb **SIMM** (**(< M SIMM)**) o bé podíem baixar la guàrdia perquè ja havíem deixat enrera aquesta incidència, en una iteració precedent en què les posicions **M** i **SIMM** actuals tenien intercanviats els noms (**(> M SIMM)**) o en la iteració **M** actual (**(> N SIMM)** perquè, en produir-se la incidència **(= N SIMM)**, **PEND** ha canviat); recordem que la coincidència revelava la incompatibilitat entre **FM** i **FN**, sense haver de recórrer a **COMPAT**. Quant a saber si era temps de sembrar o de recollir: si anomenem **SIMI** a la primera de les posicions **SIMM** i **SIMN**, i **SIMS** a la segona, **M < SIMI < SIMS** indicava que calia anar a **COMPAT** per veure si **FM** i **FN** eren o no compatibles, i guardar aquest resultat per tal que més endavant, quan comparéssim les files simètriques, els hi poguéssim atribuir la mateixa relació sense necessitat de recórrer a **COMPAT**; altrament, quan **SIMI < M** (sense importar si **M < SIMS** o **SIMS < M**), recuperàvem la informació emmagatzemada en comparar les files situades en posició **SIMI** i **SIMS**, i l'extrapolàvem a **FM** i **FN**. El vehicle per emmagatzemar i recuperar informació eren les llistes **L1SIM** i **L2SIM**, que millor seria anomenar macrollistes (de fet, pot ser que la memòria col·lapsés en ple procés de creixement d'aquestes llistes, perquè ens havíem limitat a provar un petitíssim segment inicial): la primera estava constituïda per parells puntuals **(SIMI . SIMS)** (ocupen menys que les subllistes ordinàries de 2 elements), i en la segona els elements homòlegs eren valors binaris **T** o **nil**, segons la compatibilitat de les llistes corresponents a aquestes posicions (i la de les files simètriques). I, ja sense més preàmbuls, veieu aquesta nova versió:

```
(defun C:FES-M-COMPAT-N (/ A I I1 I2 I3 J K M N FM FN LTM RLTM L1 LOC LS CC OK
                           PEND SIMM SIMN SIMI SIMS PEND L1SIM L2SIM) ; Amb SIMETR
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (CARREGA-SIMETR)
  (setq A (open "M-COMPAT-N.txt" "w") M -1 K 1)
  (while (< (setq M (1+ M)) 322560) ; 322560 = (nth 8 LOC)
         (setq SIMM (nth M LS) PEND (< M SIMM)
               FM (nth M L1)
               I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
               LTM (list I3 I2 I1) J 3)
         (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
         (setq RLTM (reverse LTM))
         (if (> I1 I2) (setq I I1 I1 I2 I2 I))
         (if (> I2 I3)
             (progn
               (setq I I2 I2 I3 I3 I)
               (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
         (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
               K (car I) CC ())
         (while (< (setq K (1+ K)) 9)
                (foreach J I (if (= K J) (setq K (1+ K))))
                (if (< K 9)
                    (progn
                      (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC))
                      (while (< (setq N (1+ N)) I1)
                             (if (and PEND (= N SIMM))
                                 (setq PEND ())
                                 (progn
                                   (setq SIMN (nth N LS) SIMI (min SIMM SIMN)
                                                         SIMS (max SIMM SIMN))
```

```
                                    (if (< M SIMI)
                                        (setq FN (nth N L1) OK (COMPAT)
                                              L1SIM (cons (cons SIMI SIMS) L1SIM)
                                              L2SIM (cons OK L2SIM))
                                        (setq OK (nth (- (length L2SIM)
                                                         (length (member (cons M N)
                                                                         L1SIM)))
                                                      L2SIM)))
                                    (if OK (setq CC (strcat (if CC CC (itoa M))
                                                            " " (itoa N))))))))))
          (if CC (write-line CC A)))
  (close A))
```

Com a les versions precedents, ens vam limitar als set primers registres però, tot i que ja prevèiem que aquesta circumstància no en permetria apreciar les possibles economies, ens va caure l'ànima als peus: de **10 minuts** per registre passavem a **20**. Sort que encara quedava el segon cartutx: **DIFA10** i l'esperança d'una intervenció més contundent en **C:FES-M-COMPAT-N**. Així que vam començar definint aquesta funció:

```
(defun DIFA10 (TXT1 / K TXT2)
  (setq TXT2 "" K 0)
  (repeat 9
    (setq K (1+ K) TXT2 (strcat TXT2 (itoa (- 10 (atoi (substr TXT1 K 1))))))))
```

Aquesta transformació tenia una particularitat que no es donava amb **SIMETR**: que la transformada d'una fila ocupava la posició simètrica en **L1**; així, si la fila **FN** se situava en la posició **N** de **L1**, la fila **(DIFA10 FN)** ocupava el lloc **(- 362879 N)**). Ho podeu comprovar per a qualsevol fila,

```
       (nth 0 L1) = "123456789" ->
                  -> "987654321" = (nth 362879 L1)
       (nth 1 L1) = "123456798" ->
                  -> "987654312" = (nth 362878 L1)
  ..............................................
  (nth 181438 L1) = "549876312" ->
                  -> "561234798" = (nth 181441 L1)
  (nth 181439 L1) = "549876321" ->
                  -> "561234789" = (nth 181440 L1)
```

però ho intentarem justificar.

En presentar **C:FES-1234567** hem dit que en el fons es tractava d'aplegar els textos formats per les **9!= 362.880** permutacions possibles dels caràcters numèrics "1"... "9", però que ho faríem creant una successió ordenada d'enters des de **123.456.789** fins a **987.654.321**, d'on s'eliminarien els valors que tinguessin algun **0** o xifres repetides, que era un procediment més fàcil. Si ara recuperem el model permutatiu i considerem les 362.880 permutacions dividides en 9 blocs de **8!= 40.320**, dintre de cadascun d'ells la xifra de l'esquerra és fixa i les 8 següents es permuten en successió creixent, podem considerar-les desglossades així:

- 2! permutacions de 2 últimes xifres: 2.
- 3! permutacions de 3 últimes xifres, menys precedents: 6 - 2 = 4.
- 4! permutacions de 4 últimes xifres, menys precedents: 24 - 6 = 18.
- 5! permutacions de 5 últimes xifres, menys precedents: 120 - 24 = 96.
- 6! permutacions de 6 últimes xifres, menys precedents: 720 - 120 = 600.
- 7! permutacions de 7 últimes xifres, menys precedents: 5.040 - 720 = 4.320.
- 8! permutacions de 8 últimes xifres, menys precedents: 40.320 - 5.040 = 35.280.

A més, aquest desglossament el podem efectuar partint del primer element i en sentit creixent, o partint de l'ultim element i en sentit decreixent. Doncs bé, si només representem els extrems d'aquest desglossament, i ho fem amb el començament del primer bloc (les 40.320 files que comencen amb "1") en sentit creixent,

```
  2! =      2: 123456789, 123456798,

  3! - 2 =  4: 123456879, 123456897, 123456978, 123456987,

  4! - 6 = 18: 123457689, 123457698, 123457869, 123457896, 123457968, 123457986,
               123458679, 123458697, 123458769, 123458796, 123458967, 123458976,
               123459678, 123459687, 123459768, 123459786, 123459867, 123459876,
  ..........................................................................
```

i la fi de l'últim (les 40.320 files que comencen amb "9") en sentit decreixent,

```
  2! =      2: 987654321, 987654312,

  3! - 2 =  4: 987654231, 987654213, 987654132, 987654123,

  4! - 6 = 18: 987653421, 987653412, 987653241, 987653214, 987653142, 987653124,
               987652431, 987652413, 987652341, 987652314, 987652143, 987652134,
               987651432, 987651423, 987651342, 987651324, 987651243, 987651234,
  ..........................................................................
```

és fàcil comprovar que els elements simètrics (homòlegs, tal com els tenim situats aquí) són complementaris de 10, i també imaginar que, si seguíssim amb la mateixa llei de formació, ho seria la resta de parells d'elements, que no hem representat. De tota manera, si algú al·legués (amb raó) que aquesta invocació a la imaginació no és una demostració rigorosa, podríem recórrer a un procediment no massa elegant però absolutament inobjectable, atès que ens referim a un conjunt discret: provar-ho per a tots els elements, mitjançant una rutina informàtica. No ha costat fer-la i l'execució sense rastre demostra que totes les parelles simètriques, convertides a valor numèric, sumen 1.111.111.110, que amb dos nombres de nou xifres 1... 9 és la forma més ràpida de manifestar que totes les seves xifres homòlogues sumen 10:

```
(defun PROVA-DIFA10 (/ M N)
  (C:CARREGA-123456789)
  (setq M -1 N 362880)
  (repeat 181440
    (setq M (1+ M) N (1- N))
    (if (/= (+ (atoi (nth M L1)) (atoi (nth N L1))) 1111111110)
        (print "\nFALLA!"))))
```

La simetria de les posicions, en la llista **L1**, d'una fila i la seva complementària de 10, ens ha fet pensar que una versió més ràpida de la funció **DIFA10** podria ser aquesta (com abans, suposem que **L1** existeix perquè ja hem carregat 123456789.txt):

```
(defun DIFA10 (TXT) (nth (- 362879 (LOCL1 TXT)) L1))
```

Doncs no: l'ús de **LOCL1** retarda considerablement el procés, de manera que la nova versió és molt més lenta que la primitiva. Un altra cosa és aprofitar el fet que la fila original i la simètrica, traduïdes a valor numèric, sumin **1.111.111.110**:

```
(defun DIFA10 (TXT) (itoa (- 1111111110 (atoi TXT))))
```

Aquesta versió, més ràpida que la primitiva, hauria estat la finalment adoptada si hagués calgut construir un arxiu de files complementàries (anàlogament al de files simètriques), mitjançant alguna funció com aquesta que hem calcat de **C:FES-SIMETR**:

```
(defun C:FES-DIFA10 (/ A M)
  (C:CARREGA-123456789)
  (setq A (open "DIFA10.txt" "w") M -1)
  (repeat 362880
    (setq M (1+ M))
    (write-line (itoa (LOCL1 (DIFA10 (nth M L1)))) A))
  (close A))
```

Anéssim a usar-la o no, en la penúltima línia no era raonable passar de la posició **M** al contingut de la fila, trobar el contingut complementari i la seva posició, i acabar transformant-la en text per escriure-la a l'arxiu, podent fer directament:

```
(defun C:FES-DIFA10 (/ A M)
  (C:CARREGA-123456789)
  (setq A (open "DIFA10.txt" "w") M -1)
  (repeat 362880
    (setq M (1+ M))
    (write-line (- 362879 M) A))
  (close A))
```

Millor encara, es podia evitar el càlcul reiterat de la diferència a 362880, fent:

```
(defun C:FES-DIFA10 (/ A M)
  (C:CARREGA-123456789)
  (setq A (open "DIFA10.txt" "w") M 362880)
  (repeat 362880
    (setq M (1- M))
    (write-line M A))
  (close A))
```

Disposant de l'arxiu DIFA10.txt, només calia una funció similar a **CARREGA-SIMETR**:

```
(defun CARREGA-DIFA10 ()
  (setq A (open "DIFA10.txt" "r"))
  (repeat 362880 (setq LD (cons (atoi (read-line A)) LD)))
  (setq LD (reverse LD))
  (close A))
```

Però, en la versió de **C:FES-M-COMPAT-N** que preteníem muntar, amb l'ajut de **DIFA10**, no calia cap llista **LD** per obtenir ràpidament la posició **(DIF M)**, fent **(nth M LD)** (lògicament, sí que necessitàvem **L1SIM** i **L2SIM**, que ara anomenàvem **L1DIF** i **L2DIF**). I és que **DIFA10** ja superava **SIMETR** jugant amb continguts, però encara més quan ho feia amb posicions a **L1**: a diferència d'aquesta funció, en què, si volíem conèixer la posició **SIMM** de la fila simètrica de **(nth M L1)** i aquesta determinació l'havíem d'efectuar reiteradament (circumstància que feia prohibitiu recórrer a l'expressió **(LOCL1 (SIMETR (nth M L1)))**), no hi havia més remei que crear prèviament l'arxiu SIMETR.txt, carregar-lo per transferir les posicions de les files simètriques a la llista **LS** i, llavors sí, fer **(setq SIMM (nth M LS))**, quan volíem saber la posició **DIFM** de la fila complementària de **(nth M L1)** n'hi havia prou a fer **(- 362879 M)**; en conseqüència, no sortia a compte crear una funció **(defun DIF (M) (- 362879 M))**. La simplicitat de la llei permetia, a més, saber fàcilment si allò que tocava fer en cada posició **M** era sembrar o recollir: com que **M < N**, en virtut de la simetria de posicions en **L1** de files complementàries també era **DIFN** < **DIFM**, de manera que només calia identificar la primera confrontació, **M** < **DIFN** (en què, anant a **COMPAT**, determinaríem si **M** i **N** eren compatibles i guardaríem les posicions **DIFN** i **DIFM** a **L1DIF** i el resultat **OK** a **L2DIF**, per utilitzar-los més endavant), distingint-la de la segona, **DIFN** < **M** (on localitzaríem **M** i **N** a **L1DIF**, i els adjudicaríem la mateixa compatibilitat que corresponia a les posicions **DIFN** i **DIFM**, vistes anteriorment i emmagatzemades a l'element homòleg de **L2DIF**). De fet, la comprovació de **M** < **DIFN** no calia efectuar-la sempre: només quan era **M ≤ 181.439** perquè, quan **181.440 ≤ M**, ja passàvem a estar en disposició permanent de collir el que havíem sembrat abans, en donar-se **DIFN < 181.440 ≤ M**; a la pràctica, el domini de **M** on calia verificar si **M** < **DIFN** encara el podíem aprimar, restringint-lo a **M < 161.280**, perquè entre **161.280 ≤ M ≤ 181.439** i **181.440 ≤ N ≤ 201.599** no es feien comparacions, començant amb "**5**" totes les files situades en aquests intèrvals. Quant a la incompatibilitat essencial entre **M** i **N** = **DIFM**, permanentment limitada al domini **M < 161.280** (fix i no pas fluctuant, com **M** < **SIMM** amb **SIMETR**), també vam usar el senyal binari **PEND**:

```
(defun C:FES-M-COMPAT-N (/ A I I1 I2 I3 J K M N FM FN LTM RLTM L1 LOC CC OK PEND
                           DIFM DIFN L1DIF L2DIF)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq A (open "M-COMPAT-N.txt" "w") M -1 K 1)
  (while (< (setq M (1+ M)) 322560) ; 322560 = (nth 8 LOC)
         (setq DIFM (- 362879 M) PEND (< M 161280)
               FM (nth M L1)
               I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
               LTM (list I3 I2 I1) J 3)
         (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
         (setq RLTM (reverse LTM))
         (if (> I1 I2) (setq I I1 I1 I2 I2 I))
         (if (> I2 I3)
             (progn
               (setq I I2 I2 I3 I3 I)
               (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
         (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
               K (car I) CC ())
         (while (< (setq K (1+ K)) 9)
                (foreach J I (if (= K J) (setq K (1+ K))))
                (if (< K 9)
```

```
                    (progn
                      (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC)
                            DIFN (- 362879 N))
                      (while (< (setq DIFN (1- DIFN) N (1+ N)) I1)
                             (if (and PEND (= N DIFM))
                                 (setq PEND ())
                                 (progn
                                   (if (< M DIFN) ; Ha de ser M < 161280
                                       (setq FN (nth N L1) OK (COMPAT)
                                             L1DIF (cons (cons DIFN DIFM) L1DIF)
                                             L2DIF (cons OK L2DIF))
                                       (setq OK (nth (- (length L2DIF)
                                                        (length (member (cons M N)
                                                                        L1DIF)))
                                                     L2DIF)))
                                   (if OK (setq CC (strcat (if CC CC (itoa M))
                                                           " " (itoa N)))))))))
        (if CC (write-line CC A)))
  (close A))
```

I encara vam poder simplificar més la funció, perquè amb l'ordenació simètrica de les files complementàries no calia que verifiquéssim si era **M** < **DIFN**, ni tan sols dintre de l'ambit restringit **M < 161.280**, perquè el moment **M** = **DIFN** ja delimitava una frontera ben clara entre la sembra (amb **PEND**) i la collita (amb **(not PEND)**):

```
(defun C:FES-M-COMPAT-N (/ A I I1 I2 I3 J K M N FM FN LTM RLTM L1 LOC CC OK PEND
                           DIFM DIFN L1DIF L2DIF)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq A (open "M-COMPAT-N.txt" "w") M -1 K 1)
  (while (< (setq M (1+ M)) 322560) ; 322560 = (nth 8 LOC)
         (setq DIFM (- 362879 M) PEND (< M 161280) FM (nth M L1)
               I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
               LTM (list I3 I2 I1) J 3)
         (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
         (setq RLTM (reverse LTM))
         (if (> I1 I2) (setq I I1 I1 I2 I2 I))
         (if (> I2 I3)
             (progn
               (setq I I2 I2 I3 I3 I)
               (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
         (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
               K (car I) CC ())
         (while (< (setq K (1+ K)) 9)
                (foreach J I (if (= K J) (setq K (1+ K))))
                (if (< K 9)
                    (progn
                      (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC)
                            DIFN (- 362879 N))
                      (while (< (setq DIFN (1- DIFN) N (1+ N)) I1)
                             (if PEND
                                 (if (= N DIFM)
                                     (setq PEND ())
                                     (setq FN (nth N L1) OK (COMPAT)
                                           L1DIF (cons (cons DIFN DIFM) L1DIF)
                                           L2DIF (cons OK L2DIF))) ; (< N DIFM)
                                 (setq OK (nth (- (length L2DIF)
                                                  (length (member (cons M N)
                                                                  L1DIF)))
                                               L2DIF))            ; (> N DIFM)
                                 (if OK (setq CC (strcat (if CC CC (itoa M)) " "
                                                         (itoa N))))))))))
         (if CC (write-line CC A)))
  (close A))
```

La durada d'aquestes versions, en què cada registre trigava **10'15"** a completar-se, de mitjana (limitant-nos, com fins ara, als set primers registres), representava una millora notable en relació als **20 minuts** per registre de la versió precedent,

en què fèiem intervenir **SIMETR**, però encara sobrepassava en 15 segons el temps de la versió primitiva de **C:FES-M-COMPAT-N**, sense "ajuts" basats en les propietats de **SIMETR** o **DIFA10**. Això tampoc no ens va agafar de sorpresa, perquè erem conscients que, de les $\mathbf{27 \times 40.320^2 = 43.893964.800}$ comparacions entre files (9 fulls enrera), de les quals era possible eludir l'accés a **COMPAT** en $\mathbf{4 \times 40.320 = 161.280}$ ocasions (sabedors que totes les files eren incompatibles amb les seves complementàries) i en la meitat de les restants (sabedors que tan compatibles com poguessin ser dues files ho eren les seves complementàries), el balanç de les **1.693.440** comparacions entre les **7** primeres files amb d'altres $\mathbf{6 \times 40.320 = 241.920}$ era poc representatiu del conjunt de comparacions i netament desfavorable, amb **1.693.412** que accedien a **COMPAT** i incrementaven les llistes **L1DIF** i **L2DIF**, i amb només **28** (set comparacions en què **(= N DIFM)** i $\mathbf{(6 + 5 + 4 + 3 + 2 + 1) = 21}$ més que recuperaven la informació d'aquestes llistes) que s'ho estalviaven: si haguéssim volgut dedicar uns quants anys (i quantitats ingents de memòria) a completar l'execució de **C:FES-M-COMPAT-N**, per fer balanç de la comparació de les files situades en posició **316.773**, **316.774**, **316.775**, **316.776**, **316.777**, **316.778** i **316.779** (les set últimes files **M** que haurien estat comparades amb les últimes 40.320 files **N**, que començaven amb "9"), hauríem vist com aquestes (fonamentalment recol·lectores) es ventilaven en molt menys de temps, en representar l'antítesi de les set primeres (fonamentalment sembradores).

Però calia ser realistes: suposant que la durada hagués quedat reduïda a la meitat (que era molt suposar, perquè la cerca a **L1DIF** i **L2DIF** també portava el seu temps) seguirien sent anys... per a una tasca preparatòria! Així que, sense voler admetre obertament que la pretensió d'exhaustivitat (totes les SUDOKUs-SOLUCIÓ) era el que duia a un carreró sense sortida, l'autor, fart perdre el temps en preparatius tan impracticables com els objectius finals (i de no tenir cap altra cosa que l'arxiu 123456789.txt), els va aparcar i va decidir que ja era inajornable disposar d'una col·lecció de tríades de files compatibles, prou assortida (tot i representar una ínfima part de les possibles), per tal d'experimentar procediments de comparació i composició de tríades, obtenint per síntesi les primeres solucions normalitzades, i de mica en mica va anar derivant cap al pla de treball exposat en el capítol 2, després d'una breu etapa d'indecisió en què simultaniejava els dos plantejaments.

D'entrada, ja es veia que la informació relativa a tríades normalitzades de files compatibles s'hauria de fragmentar, i que el més assenyat seria distribuir-la en arxius que apleguessin totes les tríades que compartien una mateixa primera fila. Acceptant que, sense cap catalogació prèvia que permetés saber quines files eren compatibles amb una de donada i limitar-se a verificar la compatibilitat interna d'aquest conjunt, la construcció de tríades de files compatibles seria força més lenta, vam emprendre aquest camí: la compatibilitat se seguiria referint a la no repetició de caràcters en els requadres 3×3; durant el procés, les files estarien representades pel contingut però, per reduir les dimensions de l'arxiu resultant, en els registres es representarien per les seva posicions a **L1**. El mateix propòsit estalviador duria a foragitar dels registres la posició de la primera fila comú, que figuraria un sol cop en el nom, precedida de "COMPAT-"; el primer camp de cada registre donaria la posició d'una segona fila compatible amb la primera (diferent en cada registre), i la resta donaria les posicions de terceres files (que podien repetir-se en altres registres), compatibles amb aquesta segona i amb la primera. També d'entrada era evident que, pretendre fer això en un sol procés ininterromput seria inviable, però encara vam pecar d'optimisme creient que per fer-ho abastable (quant a durada del procés) n'hi hauria prou a dividir-lo en sessions, referides a totes les tríades en què la primera fila comencés amb el mateix caràcter numèric. En aquesta línia, comptàvem amb què la funció **FES-COMPAT-M-N-O** s'hauria d'executar 7 vegades (amb els arguments **0**, **1**... **6**: recordem que en una tríada normalitzada de files, la primera no pot començar amb un caràcter més alt que "7"), de manera que:

- Fent **(FES-COMPAT-M-N-O 0)** crearíem els 40.320 arxius:
  COMPAT-0.txt, que contendria les compatibilitats de "123456789";
  .....
  COMPAT-40319.txt, que contendria les compatibilitats de "198765432".
- Fent **(FES-COMPAT-M-N-O 1)** crearíem els 40.320 arxius:
  COMPAT-40320.txt, que contendria les compatibilitats de "213456789";
  .....
  COMPAT-80639.txt, que contendria les compatibilitats de "298765431".

.....

- Fent **(FES-COMPAT-M-N-O 6)** crearíem els 40.320 arxius:
  COMPAT-241920.txt, que contendria les compatibilitats de "712345689";
  .....
  COMPAT-282239.txt, que contendria les compatibilitats de "798654321".

Però n'hi va haver prou amb la primera prova per copsar que encara tiràvem llarg: havent arrencat l'autor l'execució de **(FES-COMPAT-M-N-O 0)** just abans d'abandonar el despatx, deixant-la en marxa tota la nit, en tornar pel matí no el va estranyar que encara no hagués finalitzat, però sí que el va desmoralitzar el fet que ni tan sols s'hagués enllestit la construcció del primer arxiu, COMPAT-0.txt. La decepció el va dur a rebaixar els objectius, però abans de donar aquest pas mirem la funció (segueix recorrent a les mateixes funcions auxiliars **3*3** i **COMPAT**, que obviarem):

```
(defun FES-COMPAT-M-N-O (KK / RUTA A I I1 I2 I3 J K M N FM FN LTM RLT L1 L LOC CC
                             P AP MP LP)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)) M (1- (nth KK LOC)))
  (while (< (setq M (1+ M)) (nth (1+ KK) LOC))
         (setq FM (nth M L1)
               I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
               LTM (list I3 I2 I1) J 3)
         (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
         (setq RLTM (reverse LTM))
         (if (> I1 I2) (setq I I1 I1 I2 I2 I))
         (if (> I2 I3)
             (progn
               (setq I I2 I2 I3 I3 I)
               (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
         (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
               L () K KK)
         (while (< (setq K (1+ K)) 9)
                (foreach J I (if (= K J) (setq K (1+ K))))
                (if (< K 9)
                    (progn
                      (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC))
                      (while (< (setq N (1+ N)) I1)
                             (setq FN (nth N L1))
                             (if (COMPAT) (setq L (cons N L)))))))
         (if L
             (progn
               (setq A (open (strcat RUTA "COMPAT-" (itoa M) ".txt") "w")
                     L (reverse L) AP ())
               (while (and (< (setq N (car L)) 322560) ; 322560 = (nth 8 LOC)
                           (setq L (cdr L)))
                      (setq FM (nth N L1) LTM () CC () J 0)
                      (while (< J 9) (setq J (1+ J)
                                           LTM (cons (substr FM J 1) LTM)))
                      (setq RLTM (reverse LTM) P (car RLTM))
                      (if (/= P AP)
                          (progn
                            (setq J (nth (atoi P) LOC) K 0)
                            (while (and (setq K (1+ K) MP (nth K L)) (< MP J)))
                            (setq AP P)))
                      (if (setq LP (member MP L))
                          (progn
                            (foreach O LP
                                     (setq FN (nth O L1))
                              (if (COMPAT)
                                  (setq CC (strcat (if CC CC (itoa N)) " "
                                                   (itoa O)))))
                            (if CC (write-line CC A)))
                          (setq L ())))
               (close A)))))
```

Si compareu la primera meitat de **FES-COMPAT-M-N-O** amb la funció que havíem mostrat a les pàgines 32/33 (segona versió de **C:FES-M-COMPAT-N**), us adonareu de la gairebé total coincidència, incloent-hi el dispositiu per passar de llarg de les files **FN** que començaven amb algun dels tres primers caràcters de la fila **FM**, saltant-se en bloc (gràcies a la llista **LOC**) cada conjunt de 40.320 files amb tal peculiaritat. La diferència era que, en comptes d'emmagatzemar en el mateix registre d'un arxiu la posició **N** de totes les files **FN** compatibles amb **FM**, es construïa amb aquestes posicions una llista **L** que, un cop completa, era objecte d'una exploració anàloga

a l'efectuada en la primera meitat: si allà havíem comparat cada posició **M** de **L1** amb les posicions **N** superiors, en aquesta segona exploració volíem fer el mateix, recuperant fins i tot els símbols **FM**, **FN** (les seves posicions no: la del nou **FM** l'anomenàvem **N**, i la de **FN**, **O**) i algun més per poder accedir a **COMPAT**, però ara ens limitaríem a saltar-nos les **FM** que comencessin amb el primer caràcter de **FN**; el propòsit era veure si les candidates a tercera fila de la tríada (que, com la segona, eren compatibles amb la primera, en pertànyer a **L**) també eren compatibles amb la segona i, només si ho eren, guardaríem en l'arxiu les seves posicions, tal com hem dit abans: **M** figuraria en el nom de l'arxiu, després de "COMPAT-", i cada registre començaria per una posició **N** seguida per les posicions **O** corresponents a totes les files compatibles amb **FM** i **FN**, en la primera accepció d'aquests símbols.

Reprenent la narració dels fets, ens deturarem breument en un pedaç que l'autor va improvisar, impacient del tot per obtenir resultats, en un intent de flexibilitzar aquesta funció i dotar-la d'alguna utilitat pràctica: permetia de posar en marxa **FES-COMPAT-M-N-O** després d'haver-ne cancel·lat l'execució, de manera que poguéssim prosseguir la creació d'arxius just en el punt on l'havíem deixada; naturalment, calia començar creant l'arxiu que hagués quedat incomplet en la sessió precedent. A diferència de la que veurem més endavant, tenia l'avantatge de poder generar més d'un arxiu si li donàvem el temps suficient, però conceptualment era un nyap sense pal·liatius. Perquè no sigui dit, en reproduïm el començament (la resta era igual, tret del significat de **K** en l'assignació que també reproduirem), tot advertint al lector que l'argument **KK**, en comptes de representar les posicions 0, 1... 6 que en la llista **LOC** corresponien a les posicions 0, 40320... 241920 de **L1** (posicions de les files "123456789", "213456789"... "712345689"), representaven directament les posicions 0 a 282239 de **L1** (files "123456789" a "798654321") que havien de generar els arxius COMPAT-0.txt a COMPAT-282239.txt. Curiosament, i per completar el nyap, en l'assignació de **KK** a **K**, a la 20ª línia, calia substituir **KK** per la posició que ocupés en **LOC** el més elevat dels valors no superiors a l'argument **KK**. Per exemple, si preteníem executar **(FES-COMPAT-M-N-O 170000)**, abans calia modificar el codi per assignar el valor **4** a **K** (**(nth 4 LOC)** = **161280** ≤ **170000** < **201600** = **(nth 5 LOC)**):

```
(defun FES-COMPAT-M-N-O (KK / RUTA A I I1 I2 I3 J K M N FM FN LTM RLTM L1 L LOC CC
                           P AP MP LP)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)) M (1- (nth KK LOC)))
  (while (< (setq M (1+ M)) 282240) ; 282240 = (nth 7 LOC)
         (setq FM (nth M L1) ...)
         ........................
         (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
               L () K 4) ; Assigneu els valors 0, 1... 6 que corresponguin a KK.
         ..............................................................) ...)
```

Així que aquesta rutina va començar a funcionar, generant arxius COMPAT-...txt i aportant una mica de serenitat a les ànsies d'acció de l'autor, aquest es va posar a polir-la per adecentar-ne una mica l'aspecte. El primer objectiu era canviar-ne l'abast, de manera que cada execució només donés lloc a l'arxiu COMPAT-<M>.txt, determinat per l'argument **M**, en comptes d'una sèrie pràcticament ilimitada (tret d'una interrupció externa), per tal de controlar-ne millor els efectes i d'evitar perdre el temps emmagatzemant arxius incomplets que tard o d'hora caldria refer; la segona, deixar el codi tancat, de manera que l'argument **M** fos l'únic canal de tranferència de valors. Ambdues coses eren fàcils d'aconseguir: només era qüestió d'eliminar la funció **while** més exterior i de substituir els valor enter explícit, previ al tercer motor **while** (que havia passat a ser el segon), pel quocient enter **(/ M 40320)**. Quant a la nova limitació que se'ns presentava, era fàcil de superar: si l'usuari disposava de més temps del que previsiblement duraria l'execució i no volia estar pendent del final, sempre tindria la possibilitat de situar la funció, que passàvem a denominar **FES-COMPAT-3-1**, en un senzill bucle controlat per **repeat**, **foreach** o **while** (l'últim amb una caducitat més curta que **(setq M (1+ M)) 282240)**), o bé ficant-ne més d'una (amb arguments correlatius o no) dins d'una funció **progn**.

La feinada que ens esperava seguia sent ingent. Així, d'entrada, construir 282.240 arxius: tants com conjunts de tríades normalitzades que tinguessin, com a primera fila comú, cadascuna de les files compreses entre "123456789" (arxiu COMPAT-0.txt) i "798654321" (arxiu COMPAT-282239.txt). Però, a mesura que vam posar a treballar **FES-COMPAT-3-1**, vam anar veient que no en serien tants: sense anar més lluny, les últimes 11.519 files (des de "768123459" fins a "798654321") ja no trobaven files

compatibles entre les superiors, així que l'últim arxiu de la sèrie seria COMPAT-270720.txt; i no us penséssiu que només en quedaven 282.240 - 11.519 = 270.721, perquè hi ha altres llacunes, com per exemple les files "758123469" a "759864321" (arxius des de 265680 a 267119). Tot i així, n'eren un fotimer i, malgrat que la grandària era variable, podien ultrapassar els 5 MBytes. Tanmateix, era a l'hora de considerar els temps d'execució que ens queia l'ànima als peus: perquè us en feu una idea, COMPAT-240000.txt (**648 KB**) trigava a generar-se **4 hores, 40 minuts**; COMPAT-201600.txt (**1.939 KB**), **10 hores**, i COMPAT-161280.txt (**3.656 KB**), **28 hores.**

L'autor recorda com, cap al final del primer quadrimestre del curs 2007/08, durant unes setmanes, va aprofitar el fet que en el Campus Nord, en acabar la classe a les 10:55, el rellevava el Sr. J presentat en la *Introducció*, que de 11:05 a 13:55 donava una de les ALEs esmentades (crec que TÉCNICAS DE MODELADO POR ORDENADOR), amb menys alumnes, per posar en marxa els ordinadors lliures, aprofitant-los per construir alguns arxius dels previsiblement més curts (COMPAT-270715.txt a COMPAT-270718.txt, per exemple), i confiant que s'haguessin completat en tornar-hi al cap de tres hores (considerant que aquells PCs eren més ràpids que el seu del Campus Sud) i pogués copiar-se'ls en el *pendrive* abans d'anar-se'n a dinar; si no, anava a l'aula a les 15:50, un quart abans de començar la classe de tarda. Així i tot, no se li escapava que, a aquest ritme, trigaria anys i panys a completar la sèrie. Malgrat l'evidència, acabades les classes i fins que no van començar les del segon quadrimestre, en el transcurs d'aquell estrany període de temps que va de Nadal a Reis (en què els campus es buiden de personal de tota mena, quan només haurien de fer-ho d'estudiants) i durant el període que comprenia els examens, la revisió i publicació d'avaluacions i la matrícula), més d'alguna nit va deixar l'ordinador en marxa, deixant programada la creació d'arxius mitjançant expressions com ara **(progn (FES-COMPAT-3-1 201598) (FES-COMPAT-3-1 201599) (FES-COMPAT-3-1 201600))**. En tornar a l'endemà, si s'havia acabat la feina, l'apagava mitja horeta perquè es refredés, abans de baixar al bar a prendre el cafè amb llet i resoldre manualment algun dels sudokus publicats a la premsa, per espavilar-se i fer front al nou dia. Però deixem-nos de tanta jeremiada i eixuguem-nos les llàgrimes (que hauríem pogut estalviar-nos si, quatre anys abans d'escriure això, haguéssim fet bé els deures), per donar-li un cop d'ull a la funció i examinar amb cert deteniment els arxius:

```
(defun FES-COMPAT-3-1 (M / RUTA A I I1 I2 I3 J K M N FM FN LTM RLTM L1 L LOC CC P
                           AP MP LP)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq FM (nth M L1)
        I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
        LTM (list I3 I2 I1) J 3)
  (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
  (setq RLTM (reverse LTM))
  (if (> I1 I2) (setq I I1 I1 I2 I2 I))
  (if (> I2 I3)
      (progn
        (setq I I2 I2 I3 I3 I)
        (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
  (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
        K (/ M 40320))
  (while (< (setq K (1+ K)) 9)
         (foreach J I (if (= K J) (setq K (1+ K))))
         (if (< K 9)
             (progn
               (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC))
               (while (< (setq N (1+ N)) I1)
                      (setq FN (nth N L1))
                      (if (COMPAT) (setq L (cons N L)))))))
  (if L
      (progn
        (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13))
              A (open (strcat RUTA "COMPAT-" (itoa M) ".txt") "w")
              L (reverse L) AP ())
        (while (and (< (setq N (car L)) 322560) ; 322560 = (nth 8 LOC)
                    (setq L (cdr L)))
               (setq FM (nth N L1) LTM () J 0)
               (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
               (setq RLTM (reverse LTM) P (car RLTM))
```

```
            (if (/= P AP)
                (progn
                  (setq J (nth (atoi P) LOC) K 0)
                  (while (and (setq K (1+ K) MP (nth K L)) (< MP J)))
                  (setq AP P)))
            (if (setq LP (member MP L))
                (progn
                  (setq CC ())
                  (foreach O LP
                           (setq FN (nth O L1))
                           (if (COMPAT)
                               (setq CC (strcat (if CC CC (itoa N))
                                                " " (itoa O)))))
                  (if CC (write-line CC A)))
                (setq L ())))
        (close A))))
```

I just aquí s'acaba la narració dels primers passos de l'autor, fa més de quatre anys, en el camp dels sudokus. No pas perquè no en donés encara algun més, en el carreró sense sortida de l'inventari de solucions, sinó perquè el poc que va fer (com ja hem dit, de mica en mica anava desviant esforços cap a la ruta alternativa que obre el capítol 2) es basava en la funció **FES-COMPAT-3-1**, que acabem de veure, i en els arxius COMPAT-...txt que se'n derivaven. Aquest procediment de fabricació de tríades normalitzades compatibles era manifestament millorable, però l'autor no se n'ha adonat fins que ha arribat el moment de completar el guió que es va fixar, atacant el capítol 1 de l'*ÍNDEX* que, per mandra, havia anat posposant. Si recordeu una de les coses que ha confesat a l'inici d'aquest mateix capítol (que més d'una vegada, en intentar rememorar el discurs que l'havia conduït fins a una estratègia de la qual únicament conservava escrit el codi de les funcions implicades, havia descobert una errada o una alternativa millor), aquest en seria un clar exponent: com hem trobat, en repassar els vells papers, la manera de reduir l'arxiu al 4% de la seva grandària (de mitjana), treient dels registres tot allò que concernia a la tercera fila de les tríades (és a dir, eliminant tots els camps, tret del primer).

L'autor recorda vagament que es va mirar de trobar algun patró de repetició en els arxius, per tal d'eliminar informació redundant, però es va cometre l'omissió de quedar-se a l'epidermis, és a dir, en la posició de les files a **L1**, sense comparar continguts. En efecte, en un primer examen de la majoria d'aquests arxius ens pot semblar que hi ha possibilitats de reduir dràsticament la seva grandària, per la presència de conjunts de camps repetits innecessàriament, però normalment sols cal avançar una mica més per adonar-nos que les expectatives primerenques són falses. Ens referim a la impressió que, examinant ordenadament els registres, mentre els primers camps (corresponents a la 1ª fila) siguin valors correlatius, els conjunts formats per la resta (posicions de files compatibles amb la 1ª i 2ª) són idèntics, i només varien cada cop que es produeix un salt en la continuïtat del primer camp. Però, si no us detureu en els primers registres i seguiu, els contraexemples no trigaran a aparèixer: registres consecutius amb discontinuïtat entre els primers camps on, tanmateix, els conjunts formats per la resta de camps són idèntics, i registres consecutius amb els primers camps continus on, tanmateix, els conjunts formats per la resta de camps són diferents. Dissortadament, aquí s'havia quedat l'autor quan s'hi va posar per primera vegada, i no ha estat fins el moment en què li ha calgut desempolsar papers, per desxifrar les intencions ocultes darrera del codi per exposar-les aquí, que s'ha adonat que podia haver anat força més enllà: l'única possibilitat de comprimir el contingut d'aquests arxius partia de copsar que la longitud de tots els registres era de **1 + 6n** camps, perquè el conjunt de terceres files compatibles amb la primera (posició transferida al nom de l'arxiu) i la segona (posició reflectida en el primer camp del registre) estava constituït per paquets de sis camps (la posició a **L1** de sis files correlatives, idèntiques pel que fa als dos primers tercets i on cada tercer tercet era una de les **3! = 6** permutacions de tres caràcters numèrics).

Ho veureu més clar si, a títol d'exemple, reproduïm parcialment el primer registre de l'arxiu COMPAT-170000.txt i l'inici del segon, i tot seguit mostrem les files representades per aquestes posicions, destacant en negreta els 3 últims caràcters:

```
  202362 261690 261691 261692 261693 261694 261695 261714 ... 355631 355668 355669
  355670 355671 355672 355673
  202363 261690 ...
```

D'acord amb el que s'ha dit,

```
   la 1ª fila, representada per 170000 (que figura en nom_arxiu), és "528174639",
   la 2ª fila, representada per 202362 ( primer camp registre 1), és "613259478",
  una 3ª fila, representada per 261690 (següent camp registre 1), és "749368125",
  una 3ª fila, representada per 261691 (següent camp registre 1), és "749368152",
  una 3ª fila, representada per 261692 (següent camp registre 1), és "749368215",
  una 3ª fila, representada per 261693 (següent camp registre 1), és "749368251",
  una 3ª fila, representada per 261694 (següent camp registre 1), és "749368512",
  una 3ª fila, representada per 261695 (següent camp registre 1), és "749368521",
  una 3ª fila, representada per 261714 (següent camp registre 1), és "749386125",
  ..........................................................................
  una 3ª fila, representada per 355631 (següent camp registre 1), és "974836521",
  una 3ª fila, representada per 355668 (següent camp registre 1), és "974863125",
  una 3ª fila, representada per 355669 (següent camp registre 1), és "974863152",
  una 3ª fila, representada per 355670 (següent camp registre 1), és "974863215",
  una 3ª fila, representada per 355671 (següent camp registre 1), és "974863251",
  una 3ª fila, representada per 355672 (següent camp registre 1), és "974863512",
  una 3ª fila, representada per 355673 (següent camp registre 1), és "974863521",
   la 1ª fila, representada per 170000 (que figura en nom_arxiu), és "528174639",
   la 2ª fila, representada per 202363 ( primer camp registre 2), és "613259487",
  una 3ª fila, representada per 261690 (següent camp registre 2), és "749368125",
  ..........................................................................
```

Sabent això, hauríem pogut saltar-nos 5 files de **L1** cada cop que en trobéssim una de compatible amb les dues primeres (en el benentès que les files ignorades també ho eren, de compatibles) i limitar-nos a incloure aquesta posició en l'arxiu, amb la qual cosa no només hauríem reduït gairebé de cinc sisenes parts la durada de la construcció de cada arxiu sinó la seva ocupació en disc; per bé que aleshores la feina de la funció que s'hagués d'ocupar de la descàrrega de l'arxiu, transferint-ne el contingut a una llista, s'ampliaria, en haver de crear sobre la marxa cinc tríades permutades més per cadascuna de les preses de l'arxiu. Fins i tot hauríem pogut anar més lluny, en l'afany d'aprimar l'arxiu a costa de les terceres files: només cal adonar-se que el segon tercet també va repetint les **3!= 6** permutacions dels mateixos tres caràcters numèrics, i que si no passa el mateix amb el primer tercet és perquè juguem amb tríades normalitzades i cal que la 1ª fila sigui menor que la 2ª, i aquesta menor que la 3ª (circumstància que en el l'exemple exclou les permutacions 479 i 497, permetent només les 4 permutacions 749, 794, 947 i 974); per això, els registres poden constar de **1 + 2×36 = 73** camps, de **1 + 4×36 = 145** o de **1 + 6×36 = 217** (... o de **1 + 0×36 = 1**, cas en què el registre incomplet ja no s'emmagatzemaria). Comptat i debatut, hauríem pogut prescindir de tota referència a les terceres files en els arxius i quedar-nos amb registres simples (camp únic) representant les segones, perquè els tres valors de cada tercet de la tercera fila queden determinats pels dos tercets homòlegs corresponents a la primera i segona (72, 144, 216 o 0 combinacions de 3 permutacions ternàries). Doncs això és el que podem fer ara, aprofitant en bona mesura la feina realitzada fa quatre anys i mig: eliminarem la segona meitat de **FES-COMPAT-3-1**, anomenant-la **FES-KOMPAT-3-1**, i als nous arxius passarem a anomenar-los KOMPAT-...txt (bàsicament per distingir-los, ja que de moment conservarem els antics COMPAT-...txt, que ens serviran per poder comparar els continguts). Així reduïm cada registre a 1/73, 1/145 o 1/217 parts, tot i que en surten més, corresponents a files que, en realitat, no serveixen com a 2ª fila de la tríada, perquè ja no troben cap 3ª fila superior (cal recordar que treballem amb tríades normalitzades) que sigui compatible amb elles i amb la 1ª. Però d'això ja ens en ocuparem quan descarreguem aquests arxius; de moment, veiem aquesta versió retallada de **FES-COMPAT-3-1** (on **(if L ...)** passa de 25 a 5 línies):

```
(defun FES-KOMPAT-3-1 (M / RUTA A I I1 I2 I3 J K N RUTA FM FN LTM RLTM L1 L LOC)
  (C:CARREGA-123456789)
  (LOC1A9L1)
  (setq FM (nth M L1)
        I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
        LTM (list I3 I2 I1) J 3)
  (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
  (setq RLTM (reverse LTM))
  (if (> I1 I2) (setq I I1 I1 I2 I2 I))
  (if (> I2 I3)
      (progn
        (setq I I2 I2 I3 I3 I)
        (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
```

```
  (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
        K (/ M 40320))
  (while (< (setq K (1+ K)) 9)
         (foreach J I (if (= K J) (setq K (1+ K))))
         (if (< K 9)
             (progn
               (setq N (1- (nth K LOC)) I1 (nth (1+ K) LOC))
               (while (< (setq N (1+ N)) I1)
                      (setq FN (nth N L1))
                      (if (COMPAT) (setq L (cons N L)))))))
  (if L
      (progn
        (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13))
              A (open (strcat RUTA "KOMPAT-" (itoa M) ".txt") "w"))
        (foreach E (reverse L) (write-line (itoa E) A))
        (close A))))
```

Encara podem alleugerir **FES-KOMPAT-3-1** prescindint de l'accés a la funció **LOC1A9L1** (de fet, com que ja no usarem més la llista **LOC**, ens podríem oblidar de la funció **LOC1A9L1**) i canviant el significat de **K**, que passaria de representar el caràcter inicial de **FM** (valor numèric) a la posició de la primera fila amb aquest caràcter:

```
(defun FES-KOMPAT-3-1 (M / RUTA A I I1 I2 I3 J K N RUTA FM FN LTM RLTM L1 L)
  (C:CARREGA-123456789)
  (setq FM (nth M L1)
        I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
        LTM (list I3 I2 I1) J 3)
  (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
  (setq RLTM (reverse LTM))
  (if (> I1 I2) (setq I I1 I1 I2 I2 I))
  (if (> I2 I3)
      (progn
        (setq I I2 I2 I3 I3 I)
        (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
  (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
        K (* (/ M 40320) 40320))
  (while (< (setq K (+ K 40320)) 362880)
         (foreach J I (if (= K (* J 40320)) (setq K (+ K 40320))))
         (if (< K 362880)
             (progn
               (setq N (1- K) I1 (princ (+ K 40320)))
               (while (< (setq N (1+ N)) I1)
                      (setq FN (nth N L1))
                      (if (COMPAT) (setq L (cons N L)))))))
  (if L
      (progn
        (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13))
              A (open (strcat RUTA "KOMPAT-" (itoa M) ".txt") "w"))
        (foreach E (reverse L) (write-line (itoa E) A))
        (close A))))
```

Tornem ara als registres sobrers, corresponents a segones-files-polissó i que, amb força paciència, podríem localitzar comparant versions homòlogues KOMPAT i COMPAT. Per tenir una idea del volum de passatge clandestí que es pot infiltrar a l'arxiu, que sovint supera en nombre al legal, prendrem com a referència KOMPAT-240000.txt, amb un total de 4.032 segones files enregistrades de les quals només n'hi ha 1.296 d'aprofitables: de les 1.620 primeres (posicions 242076 a 259199), només 1.296 són passatgers amb bitllet, perquè hi ha 324 polissons barrejats; de les 2412 restants (posicions 261090 a 315359) totes són polissons; així doncs, més endavant caldrà desprendre's de 2.736 segones files inútils, perquè si cal complir les convencions de les tríades normalitzades no trobaran cap tercera fila compatible. Malgrat tot, l'ocupació dels arxius minva espectacularment (exemples: COMPAT-240000.txt ocupava **648 KB**, mentre que KOMPAT-240000.txt n'ocupa **32**; COMPAT-201600.txt ocupava **1.939 KB**, mentre que KOMPAT-201600.txt n'ocupa **48**; COMPAT-131280.txt ocupava **3.655 KB**, mentre que KOMPAT-131280.txt n'ocupa **63**), i els temps de processat encara més (per crear els arxius anteriors, passem respectivament de **4 hores amb 40'** a **4'15"**, de **10 hores** a **7'10"** i de **28 hores** a **8'45"**).

Aquest salt qualitatiu permet plantejar-se la consecució d'una funció **FES-SUDOKUS** capaç de generar múltiples SUDOKUs-SOLUCIÓ a partir de tres arxius KOMPAT-...txt, que haurà de comptar amb una funció auxiliar **CARREGA-TRIADES** per transferir-ne els continguts a sengles llistes. Com que aquests arxius ja estan reduïts a la mínima expressió, en justa contrapartida **CARREGA-TRIADES** no sols capturarà informació per interpretar-la (passant de les posicions als continguts **TF1** i **TF2** dels elements de **L1**) sinó per ampliar-la, generant totes les terceres files **TF3** compatibles amb la 1ª i 2ª. En la tasca d'identificar en cada cada columna els 3 caràcters numèrics pendents, mitjançant la funció **RESTA**, i de formar totes les permutacions d'aquests valors en cada tercet, mitjançant **PERM**, aquesta última funció s'encarregarà també de descartar aquelles que corresponguin a files situades a **L1** abans de la 1ª i 2ª: no li cal esperar a tenir els 3 tercets per constatar aquesta eventualitat, perquè n'hi ha prou a mirar si el caràcter inicial del primer tercet és inferior al de la segona fila. Si de resultes d'aquesta selecció no quedés cap 3ª fila, això voldria dir que la segona fila era un dels polissons a què fèiem referència en el paràgraf precedent, i en consequència no aportaria cap tríada a la llista **LTF123** resultant. Així doncs, en aquesta llista només hi apareixen les segones files viables i les terceres a què donen lloc (sense oblidar la primera, la posició **M** de la qual serà en valor subministrat a **CARREGA-TRIADES**). Aclarint que **LTF123** tindrà l'estructura **(TF1 (TF2 (TF3... TF3)) (TF2 (TF3... TF3))... (TF2 (TF3... TF3)))**, veieu el codi:

```
(defun RESTA (T1 T2 / T21 LC)
  (setq T21 (strcat T2 T1))
  (foreach C C9A1      ; C9A1 = '("9" "8" "7" "6" "5" "4" "3" "2" "1")
    (if (not (wcmatch T21 (strcat "*" C "*"))) (setq LC (cons C LC))))
  LC)

(defun CAT (C1 C2 C3) (strcat (nth C1 LC) (nth C2 LC) (nth C3 LC)))

(defun PERM (LC COMP / LT3 LT OK)
  (setq LT3 (list (CAT 0 1 2))
        LT3 (cons (CAT 0 2 1) LT3)
        LT3 (cons (CAT 1 0 2) LT3)
        LT3 (cons (CAT 1 2 0) LT3)
        LT3 (cons (CAT 2 0 1) LT3)
        LT3 (cons (CAT 2 1 0) LT3))
  (if COMP
      (progn
        (setq OK T)
        (foreach T3 LT3
          (if OK
              (if (< (substr T3 1 1) COMP)
                  (setq OK ())
                  (setq LT (cons T3 LT)))))
         LT)
      (reverse LT3)))

(defun CARREGA-TRIADES (A M / TF1-1 TF1-2 TF1-3 TF2-1 TF2-2 TF2-3 TF3-1 TF3-2
                              TF3-3 LC3-1 LC3-2 LC3-3 LTF3 LTF23 LTF123)
  (setq A (open A "r") TF1 (nth M L1)
        TF1-1 (substr TF1 1 3) TF1-2 (substr TF1 4 3) TF1-3 (substr TF1 7 3))
  (while (setq TF2 (read-line A))
         (setq TF2 (nth (atoi TF2) L1) TF2-1 (substr TF2 1 3)
               LC3-1 (RESTA TF1-1 TF2-1) LTF3-1 (PERM LC3-1 (substr TF2-1 1 1)))
         (if LTF3-1
             (progn
               (setq TF2-2 (substr TF2 4 3) TF2-3 (substr TF2 7 3)
                     LC3-2 (RESTA TF1-2 TF2-2) LTF3-2 (PERM LC3-2 ())
                     LC3-3 (RESTA TF1-3 TF2-3) LTF3-3 (PERM LC3-3 ())
                     LTF23 () LTF3 ())
               (foreach TF3-1 LTF3-1
                 (foreach TF3-2 LTF3-2
                   (foreach TF3-3 LTF3-3 ; No ens cal TF3
                     (setq LTF3 (cons (strcat TF3-1 TF3-2 TF3-3) LTF3)))))
               (setq LTF23 (list TF2 (reverse LTF3))
                     LTF123 (cons LTF23 LTF123)))))
  (close A)
  (cons TF1 (reverse LTF123)))
```

¿Per què obliguem l'usuari a subministrar com a argument addicional el nom **A** de l'arxiu KOMPAT-<posició_a_L1_de_la_1ª_fila>.txt, quan és fàcilment deduïble de **M**, i per què no hem declarat les variables **TF1**, **TF2** i **TF3**? Doncs perquè el destí de **CARREGA-TRIADES** no és ser executada explícitament per l'usuari sinó esdevenir una funció auxiliar de **FES-SUDOKUS**, i en aquest context aprofitarem les variables **TF1**, **TF2** i **TF3**, locals en aquesta funció, per representar la 1ª, 2ª i 3ª fila de cada tríada normalitzada (a **FES-SUDOKUS** tindran un significat diferent, tret de **TF1** en el primer dels tres accessos a **CARREGA-TRIADES**), però abans d'entrar en la lletra menuda tornem a la línia discursiva principal veient la raó de ser de **FES-SUDOKUS**.

Si, per no fer-ho tan llarg com KOMPAT-<posició_a_L1_de_la_1ª_fila>.txt, convenim d'anomenar KOMPAT-<pF> l'arxiu de totes les tríades que comparteixen una primera fila **F** que té la posició <pF> a **L1** (notació alternativa a fila **FM**, de posició **M**), reservant els nombres ordinals per referir-nos a la posició que ocuparan aquestes primeres files en una SUDOKU-SOLUCIÓ concebuda com a juxtaposició de tres tríades compatibles entre si, direm que l'objectiu de la funció **FES-SUDOKUS** és inventariar les SUDOKUs-SOLUCIÓ que puguin formar-se amb les tríades dels arxius KOMPAT-<pF1>, KOMPAT-<pF4> i KOMPAT-<pF7>, donant lloc a un arxiu de Solucions Normalitzades que portarà el nom SN-<pF1>-<pF4>-<pF7> (també amb extensió .txt). Naturalment, abans d'executar **FES-SUDOKUS** caldrà que haguem construït els tres arxius de tríades (amb la funció **FES-KOMPAT-3-1**), arxius que hauran de complir els requisits següents:
- **0 ≤** <pF1> **≤ 40.319** perquè **F1** ha de començar amb el caràcter **"1"**.
- **40.320 ≤** <pF4> **≤ 161.279** perquè el primer caràcter de **F4** serà **"2"**, **"3"** o **"4"**:
  - Si és **"2"**: **80.640 ≤** <pF7> perquè el primer caràcter de **F7** ha de ser superior.
  - Si és **"3"**: **120.960 ≤** <pF7> perquè el primer caràcter de **F7** ha de ser superior.
  - Si és **"4"**: **161.280 ≤** <pF7> perquè el primer caràcter de **F7** ha de ser superior.
- <pF7> **≤ 282.239** perquè el primer caràcter de **F7** no ha de ser superior a **"7"**.

Aquestes condicions (a les quals caldrà afegir que, d'entrada, la fila **F4** no sigui incompatible amb **F1**, i que la fila **F7** no sigui incompatible amb **F1** ni amb **F4**, en el sentit més ampli de no haver-hi coincidències en columna), seran exigides quan se li demanin a l'usuari les posicions <pF1>, <pF4> i <pF7>, i el seu incompliment donarà lloc al rebuig l'entrada que pertoqui.

I, pel que fa a la determinació de la compatibilitat (també en el sentit més ampli de no haver-hi coincidències en columna), a ningú se li acudiria tirar pel dret i posar-se a comparar-ho tot amb tot, a l'estil del que podria ser aquest algorisme:

```
(setq LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
      LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2)
      LT3 (CARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3)
      A3 (open (strcat RUTA "SN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
               "w"))
(foreach T1F23 LT1F23
  (setq T1F2 (car T1F23) LT1F3 (cadr T1F23))
  (foreach T1F3 LT1F3
    (foreach T2F23 LT2F23
      (setq T2F2 (car T2F23) LT2F3 (cadr T2F23))
      (foreach T2F3 LT2F3
        (foreach T3F23 LT3F23
          (setq T3F2 (car T3F23) LT3F3 (cadr T3F23))
          (foreach T3F3 LT3F3
            (setq K 0 OK T)
            (while (and OK (< (setq K (1+ K)) 10))
                   (setq COL (list (substr T1F1 K 1) (substr T1F2 K 1)
                                   (substr T1F3 K 1) (substr T2F1 K 1)
                                   (substr T2F2 K 1) (substr T2F3 K 1)
                                   (substr T3F1 K 1) (substr T3F2 K 1)
                                   (substr T3F3 K 1)))
                   (foreach J '("1" "2" "3" "4" "5" "6" "7" "8" "9")
                     (if OK (setq OK (member J COL)))))
            (if OK (write-line
                     (strcat (itoa (LOCL1 T1F2)) " " (itoa (LOCL1 T1F3)) " "
                             (itoa (LOCL1 T2F2)) " " (itoa (LOCL1 T2F3)) " "
                             (itoa (LOCL1 T3F2)) " " (itoa (LOCL1 T3F3)))
                    A3))))))))
(close A3)
```

Actuar tan barroerament no només implicaria, per exemple, que una incompatibilitat entre la 1ª, 2ª o 3ª fila i la 5ª (2ª fila de la segona tríada) hauria d'esperar al final per ser detectada (és a dir, caldria capturar la primera de les sisenes, la primera de les vuitenes i la primera de les novenes per poder constatar aquesta circumstància, en què les tres últimes no hi estaven involucrades), sinó que això no hauria tingut cap conseqüència als efectes d'evitar una munió de comparacions inútils (amb la resta de les sisenes, la resta de les vuitenes i la resta de les novenes). Per això és imperatiu no deixar-ho tot per al final i procedir de forma gradual, recolzant-nos amb aquesta sèrie de consideracions, cadascuna de les quals aplicarem quan correspongui:

- 2ª fila tríada 1 ha de ser compatible amb 1ª fila tríada 2 i 1ª fila tríada 3 (amb 1ª fila tríada 1 ja sabem que ho és).
- 3ª fila tríada 1 ha de ser compatible amb 1ª fila tríada 2 i 1ª fila tríada 3 (amb 1ª i 2ª files tríada 1 ja sabem que ho és).
- 2ª fila tríada 2 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª fila tríada 3 (amb 1ª fila tríada 2 ja sabem que ho és).
- 3ª fila tríada 2 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª fila tríada 3 (amb 1ª i 2ª files tríada 2 ja sabem que ho és).
- 2ª fila tríada 3 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª, 2ª i 3ª fila tríada 2 (amb 1ª fila tríada 3 ja sabem que ho és).
- 3ª fila tríada 3 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª, 2ª i 3ª fila tríada 2 (amb 1ª i 2ª files tríada 3 ja sabem que ho és).

Aquestes condicions es materialitzaran utilitzant la funció predefinida **wcmatch**, que aplicarem a les 6 files esmentades (**T1F2** i **T1F3** són la 2ª i 3ª de la tríada 1; **T2F2** i **T2F3** són la 2ª i 3ª de la tríada 2; **T3F2** i **T3F3** són la 2ª i 3ª de la tríada 3), comparant les dues primeres amb el perfil **PERF1** (cap caràcter no pot coincidir amb els de la 4ª o 7ª fila -**T2F1** o **T3F1**- que se situïn a la mateixa columna), les dues següents amb **PERF2** (cap caràcter no pot coincidir amb els de la 1ª, 2ª, 3ª o 7ª fila -**T1F1**, **T1F2**, **T1F3** o **T3F1**- que se situïn a la mateixa columna) i les dues últimes amb **PERF3** (cap caràcter no pot coincidir amb els de la 1ª, 2ª, 3ª, 4ª, 5ª o 6ª fila - **T1F1**, **T1F2**, **T1F3**, **T2F1**, **T2F2** o **T2F3**- situats en la mateixa columna). Els perfils es compondran mitjançant la funció **COMP**: aquesta funció no té res a veure amb **COMPAT**, i si us ha de produir maldecaps pensar que els perfils **wcmatch** també són patrons de compatibilitat (en aquest cas caràcter a caràcter, és a dir en el sentit més ampli de no haver-hi coincidències en columna), us podeu imaginar que el nom es refereix a la **comp**osició del perfil de text, no a la **comp**atibilitat. En definitiva, **FES-SUDOKUS** constarà de tres parts: la primera per a la introducció de dades (posicions de la 1ª, 4ª i 7ª fila en **L1**), la segona per descarregar el contingut dels tres arxius KOMPAT-... implicats, ampliat a les terceres files de la tríada i donant lloc a les llistes **LT1**, **LT2** i **LT3** (així que en tinguem una i desglossem el primer element de la resta, podríem anul·lar-la i alliberar memòria, ja que amb aquestes llistes tan llargues estem fregant els límits d'emmagatzament, però no ho farem fins més endavant, quan un col·lapse ens hi obligui), i l'última obrirà el sistema de sis bucles, niats i subordinats a les condicions enumerades, on s'acabarà determinant quines combinacions de 2ª, 3ª, 5ª, 6ª, 8ª i 9ª files són compatibles entre si i amb la 1ª, 4ª i 7ª, i han de ser enregistrades en un arxiu:

```
(defun BLANCS (A B / CC)
  (setq CC "")
  (repeat (- (strlen (itoa A)) (strlen (itoa B))) (setq CC (strcat CC " ")))
  (if CC CC ""))

(defun COMP (LF / PERF P)
  (setq K 0 PERF "")
  (repeat 9
    (setq K (1+ K) P "")
    (foreach F LF (setq P (strcat P (substr F K 1))))
    (setq PERF (strcat PERF "[~" P "]"))))

(defun FES-SUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1 OK2
                      LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1 LT3F23 PERF1 T1F2
                      LT1F3 PERF2 T2F2 LT2F3 PERF3 T3F2 LT3F3)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)) A1 T A2 T A3 T
        C9A1 '("9" "8" "7" "6" "5" "4" "3" "2" "1"))
```

```
(while (progn
         (setq M1 (fix (getreal (strcat "\n.\nEntreu el nom d'algun arxiu de "
                                        "tríades compatibles existent i que "
                                        "correspongui\na la primera tríada del "
                                        "SUDOKU-SOLUCIÓ (de KOMPAT-0.txt a "
                                        "KOMPAT-40319.txt),\nsense extensió "
                                        ".txt: KOMPAT-"))))
         (not (and (>= M1 0) (<= M1 40319)
                   (setq A1 (findfile (strcat RUTA "KOMPAT-" (itoa M1)
                                              ".txt"))))))
       (if (not A1)
           (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no hi "
                           "ha cap arxiu KOMPAT-" (itoa M1) ".txt."))
           (prompt (strcat "\n.\nEl valor introduït se situa fora dels límits "
                           "assenyalats.")))
       (prompt "\n                        [Per repetir, polseu qualsevol tecla]")
       (grread))
(setq TF1 (nth M1 L1))
(while (progn
         (setq M2 (fix (getreal (strcat "\n.\nEntreu el nom d'algun arxiu de "
                                        "tríades compatibles existent i que "
                                        "correspongui\na la segona tríada del "
                                        "SUDOKU-SOLUCIÓ (KOMPAT-40320.txt a "
                                        "KOMPAT-161279.txt),\nsense extensió "
                                        ".txt: KOMPAT-")))
               TF2 (nth M2 L1))
         (not (and (>= M2 40320) (<= M2 161279)
                   (setq K 0 OK T A2 (findfile (strcat RUTA "KOMPAT-" (itoa M2)
                                                       ".txt")))
                   (progn
                     (repeat 9
                        (if OK (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                            (substr TF2 K 1)))))
                      OK))))
       (if (not A2)
           (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no hi "
                           "ha  cap arxiu KOMPAT-" (itoa M2) ".txt."))
           (if OK
               (prompt (strcat "\n.\nEl valor introduït se situa fora dels "
                               "límits assenyalats."))
               (prompt (strcat "\n.\nLa fila " (itoa M2) "ª (" TF2 ") és "
                               "incompatible\n" (BLANCS M2 M1) " amb la "
                               (itoa M1) "ª (" TF1 ")."))))
       (prompt "\n                        [Per repetir, polseu qualsevol tecla]")
       (grread))
(setq LIMI (if (= (substr TF2 1 1) "2")
               "80640"
               (if (= (substr TF2 1 1) "3") "120960" "161280")))
(while (progn
         (setq M3 (fix (getreal (strcat "\n.\nEntreu el nom d'algun arxiu de "
                                        "tríades compatibles existent i que "
                                        "correspongui\na la tercera tríada del "
                                        "SUDOKU-SOLUCIÓ (KOMPAT-" LIMI
                                        ".txt a KOMPAT-282239.txt),"
                                        "\nsense extensió .txt: KOMPAT-"))))
         (not (and (>= M3 (atoi LIMI)) (<= M3 282239)
                   (setq A3 (findfile (strcat RUTA "KOMPAT-" (itoa M3) ".txt")))
                   (progn
                     (setq TF3 (nth M3 L1) K 0 OK1 T OK2 T)
                     (repeat 9 (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                            (substr TF3 K 1))
                                      OK1 (if OK OK1 ())
                                      OK (/= (substr TF2 K 1) (substr TF3 K 1))
                                      OK2 (if OK OK2 ())))
                     (setq OK (and OK1 OK2))))))
       (if (not A3)
           (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no hi "
                           "ha cap arxiu KOMPAT-" (itoa M3) ".txt."))
```

```
            (if OK
                (prompt (strcat "\n.\nEl valor introduït se situa fora dels "
                                "límits assenyalats."))
                (prompt (strcat "\n.\nLa fila " (itoa M3) "ª (" TF3 ") és "
                                "incompatible\n"
                                (if (not OK2)
                                    (strcat (BLANCS M3 M2) " amb la " (itoa M2)
                                            "ª (" TF2 ")" (if OK1 "." " i\n"))
                                    "")
                                (if (not OK1)
                                    (strcat (BLANCS M3 M1) " amb la " (itoa M1)
                                            "ª (" TF1 ").")
                                    ""))))))
        (prompt "\n                    [Per repetir, polseu qualsevol tecla]")
        (grread))
  (setq LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
        LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2)
        LT3 (CARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3)
        PERF1 (COMP (list T2F1 T3F1))
        A3 (open (strcat RUTA "SN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
                 "w"))
  (foreach T1F23 LT1F23
    (setq T1F2 (car T1F23))
; 2ª fila tríada 1 ha de ser compatible amb 1ª fila tríada 2 i 1ª fila tríada 3.
    (if (wcmatch T1F2 PERF1)
        (progn
          (setq LT1F3 (cadr T1F23))
          (foreach T1F3 LT1F3
; 3ª fila tríada 1 ha de ser compatible amb 1ª fila tríada 2 i 1ª fila tríada 3.
            (if (wcmatch T1F3 PERF1)
                (progn
                  (setq PERF2 (COMP (list T1F1 T1F2 T1F3 T3F1)))
                  (foreach T2F23 LT2F23
                    (setq T2F2 (car T2F23))
; 2ª fila tríada 2 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª
; fila tríada 3.
                    (if (wcmatch T2F2 PERF2)
                        (progn
                          (setq LT2F3 (cadr T2F23))
                          (foreach T2F3 LT2F3
; 3ª fila tríada 2 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª
; fila tríada 3.
                            (if (wcmatch T2F3 PERF2)
                                (progn
                                  (setq PERF3 (COMP (list T1F1 T1F2 T1F3 T2F1 T2F2
                                                          T2F3)))
                                  (foreach T3F23 LT3F23
                                    (setq T3F2 (car T3F23))
; 2ª fila tríada 3 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª,
; 2ª i 3ª fila tríada 2.
                                    (if (wcmatch T3F2 PERF3)
                                        (progn
                                          (setq LT3F3 (cadr T3F23))
                                          (foreach T3F3 LT3F3
; 3ª fila tríada 3 ha de ser compatible amb 1ª, 2ª i 3ª fila tríada 1, i amb 1ª,
; 2ª i 3ª fila tríada 2.
                                            (if (wcmatch T3F3 PERF3)
                                                (write-line
                                                  (strcat (itoa (LOCL1 T1F2)) " "
                                                          (itoa (LOCL1 T1F3)) " "
                                                          (itoa (LOCL1 T2F2)) " "
                                                          (itoa (LOCL1 T2F3)) " "
                                                          (itoa (LOCL1 T3F2)) " "
                                                          (itoa (LOCL1 T3F3)))
                                                  A3)))))))))))))))))
  (close A3)
  (princ))
```

El resultat, però, ha estat decebedor: prenent com a referència la construcció de l'arxiu SN-39600-46233-92464.txt, a partir dels 3 arxius de tríades normalitzades compatibles KOMPAT-39600.txt, KOMPAT-46233.txt i KOMPAT-92464.txt (més endavant ja veureu d'on surt l'exemple), dels 255.091 registres-solució de què consta l'arxiu complet (també ho veureu més endavant), en 25 minuts només se n'han creat **3.680.** Les pròximes nou pàgines són una exposició de les variacions ideades sobre aquest tema amb la intenció de superar àmpliament un rècord que benèvolament qualificarem de discret: cal reconèixer que hi ha hagut més bones intencions que no pas avenços significatius; molts han estat irrellevants i alguns s'han traduït en retrocessos.

El primer intent va consistir en l'intent de disminuir l'accés a **wcmatch**, apartant prèviament aquelles files que començaven amb un caràcter inferior al més baix dels caràcters que encara no havien aparegut a la primera columna. Per fer això: anàvem actualitzant el text constituït pels primers caràcters de les files existents, que prenia el nom **CC12**, **CC13**, **CC15**, **CC16**, **CC18** o **CC19**, segons que entiguéssim pendents d'introduir la 2ª, 3ª, 5ª, 6ª, 8ª o 9ª fila; del complement d'aquests textos (és a dir, dels caràcters "1", "2"... "9" que no hi sortien, que obteníem aprofitant la funció **RESTA**) en seleccionàvem l'inferior, que adoptava els noms **C1INF2**, **C1INF3**, **C1INF5**, **C1INF6**, **C1INF8** i **C1INF9**; limitant-nos al primer, substituíem la condició **(if (wcmatch T1F2 PERF1)...** per **(if (and (> T1F2 C1INF2) (wcmatch T1F2 PERF1))...** (fixeu-vos que utilitzem la comparació **>** i no pas la **≥** perquè, quan la fila **T1F2** de 9 caràcters comenci amb el caràcter **C1INF2**, la relació que els lligarà serà la primera). De les tres parts que abans consideràvem en **FES-SUDOKU**, ens limitarem a presentar l'encapçalament i la 2ª i 3ª que, quant a extensió del codi, representa més o menys la segona meitat (a partir d'ara, ens hi referirem d'aquesta manera):

```
(defun FES-SUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1 OK2
                      LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1 LT3F23 CC12 C1INF2
                      PERF1 T1F2 LT1F3 CC13 C1INF3 CC15 C1INF5 PERF2 T2F2 LT2F3
                      CC16 C1INF6 CC18 C1INF8 PERF3 T3F2 LT3F3 CC19 C1INF9)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  ................................................................
  (setq LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
        LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2)
        LT3 (CARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3)
        CC12 (strcat (substr T1F1 1 1) (substr T2F1 1 1) (substr T3F1 1 1))
        C1INF2 (car (RESTA CC12 "")) PERF1 (COMP (list T2F1 T3F1))
        A3 (open (strcat RUTA "SN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
                 "w"))
  (foreach T1F23 LT1F23
    (setq T1F2 (car T1F23))
    (if (and (> T1F2 C1INF2) (wcmatch T1F2 PERF1))
        (progn
          (setq LT1F3 (cadr T1F23)
                CC13 (strcat CC12 (substr T1F2 1 1))
                C1INF3 (car (RESTA CC13 "")))
          (foreach T1F3 LT1F3
            (if (and (> T1F3 C1INF3) (wcmatch T1F3 PERF1))
                (progn
                  (setq CC15 (strcat CC13 (substr T1F3 1 1))
                        C1INF5 (car (RESTA CC15 ""))
                        PERF2 (COMP (list T1F1 T1F2 T1F3 T3F1)))
                  (foreach T2F23 LT2F23
                    (setq T2F2 (car T2F23))
                    (if (and (> T2F2 C1INF5) (wcmatch T2F2 PERF2))
                        (progn
                          (setq LT2F3 (cadr T2F23)
                                CC16 (strcat CC15 (substr T2F2 1 1))
                                C1INF6 (car (RESTA CC16 "")))
                          (foreach T2F3 LT2F3
                            (if (and (> T2F3 C1INF6) (wcmatch T2F3 PERF2))
                                (progn
                                  (setq CC18 (strcat CC16 (substr T2F3 1 1))
                                        C1INF8 (car (RESTA CC18 ""))
                                        PERF3 (COMP (list T1F1 T1F2 T1F3 T2F1 T2F2
                                                          T2F3)))
```

```
                                    (foreach T3F23 LT3F23
                                      (setq T3F2 (car T3F23))
                                      (if (and (> T3F2 C1INF8) (wcmatch T3F2 PERF3))
                                          (progn
                                            (setq LT3F3 (cadr T3F23)
                                                  CC19 (strcat CC18
                                                               (substr T3F2 1 1))
                                                  C1INF9 (car (RESTA CC19 "")))
                                            (foreach T3F3 LT3F3
                                              (if (and (> T3F3 C1INF9)
                                                       (wcmatch T3F3 PERF3))
                                                  (write-line
                                                    (strcat (itoa (LOCL1 T1F2)) " "
                                                            (itoa (LOCL1 T1F3)) " "
                                                            (itoa (LOCL1 T2F2)) " "
                                                            (itoa (LOCL1 T2F3)) " "
                                                            (itoa (LOCL1 T3F2)) " "
                                                            (itoa (LOCL1 T3F3)))
                                                    A3))))))))))))))))))
  (close A3)
  (princ))
```

La pretesa optimització no només es va revelar inoperant sinò entorpidora (en els mateixos 25 minuts, únicament es van crear **3.640** registres, per sota de la versió inicial). Això, que ens ha pasat en més d'una ocasió i que tornarà a passar, no va obstar perquè insistíssim en una línia que en teoria havia de reportar avantatges. Amb petits additaments (aprofitar la variable binària **OK2**, i crear **OK3**, **OK5**, **OK6**, **OK8** i **OK9**) vam intentar d'optimitzar aquest dispositiu, fent que **(> T1F2 C1INF2)** no tornés a ser avaluada, una vegada hagués resultat **T**; que **(> T1F3 C1INF3)** no tornés a ser avaluada, una vegada hagués resultat **T**, mentre no canviés **T1F2**; que **(> T2F2 C1INF5)** no tornés a ser avaluada, una vegada hagués resultat **T**, mentre no canviés **T1F3**; que **(> T2F3 C1INF6)** no tornés a ser avaluada, un cop hagués resultat **T**, mentre no canviés **T2F2**; que **(> T3F2 C1INF8)** no tornés a ser avaluada, un cop hagués resultat **T**, mentre no canviés **T2F3**, i que **(> T3F3 C1INF9)** no tornés a ser avaluada, un cop hagués resultat **T**, mentre no canviés **T3F2**. Aquestes omissions ens les podiem permetre, perquè en les llistes **LT1**, **LT2** i **LT3** les segones i terceres files estaven ordenades de forma ascendent, però la recuperació de la pèrdua només va ser parcial (en 25 minuts, es van crear **3.660** registres, encara per sota de la primera versió). Malgrat això, us en mostrarem l'encapçalament i la segona meitat, perquè veieu la disposició de les variables i entengueu millor quin paper juguen:

```
(defun FES-SUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1 OK2
                      OK3 OK5 OK6 OK8 OK9 LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1
                      LT3F23 CC12 C1INF2 PERF1 T1F2 LT1F3 CC13 C1INF3 CC15 C1INF5
                      PERF2 T2F2 LT2F3 CC16 C1INF6 CC18 C1INF8 PERF3 T3F2 LT3F3
                      CC19 C1INF9)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  ..............................................................
  (setq LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
        LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2)
        LT3 (CARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3)
        CC12 (strcat (substr T1F1 1 1) (substr T2F1 1 1) (substr T3F1 1 1))
        C1INF2 (car (RESTA CC12 ""))
        PERF1 (COMP (list T2F1 T3F1) OK2 ())
        A3 (open (strcat RUTA "SN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
                 "w"))
  (foreach T1F23 LT1F23
    (setq T1F2 (car T1F23))
    (if (and (or OK2 (setq OK2 (> T1F2 C1INF2))) (wcmatch T1F2 PERF1))
        (progn
          (setq LT1F3 (cadr T1F23)
                CC13 (strcat CC12 (substr T1F2 1 1))
                C1INF3 (car (RESTA CC13 "")) OK3 ())
          (foreach T1F3 LT1F3
            (if (and (or OK3 (setq OK3 (> T1F3 C1INF3))) (wcmatch T1F3 PERF1))
                (progn
```

```
                    (setq CC15 (strcat CC13 (substr T1F3 1 1))
                          C1INF5 (car (RESTA CC15 ""))
                          PERF2 (COMP (list T1F1 T1F2 T1F3 T3F1)) OK5 ())
                    (foreach T2F23 LT2F23
                      (setq T2F2 (car T2F23))
                      (if (and (or OK5 (setq OK5 (> T2F2 C1INF5)))
                               (wcmatch T2F2 PERF2))
                          (progn
                            (setq LT2F3 (cadr T2F23)
                                  CC16 (strcat CC15 (substr T2F2 1 1))
                                  C1INF6 (car (RESTA CC16 "")) OK6 ())
                            (foreach T2F3 LT2F3
                              (if (and (or OK6 (setq OK6 (> T2F3 C1INF6)))
                                       (wcmatch T2F3 PERF2))
                                  (progn
                                    (setq CC18 (strcat CC16 (substr T2F3 1 1))
                                          C1INF8 (car (RESTA CC18 ""))
                                          PERF3 (COMP (list T1F1 T1F2 T1F3 T2F1 T2F2
                                                            T2F3))
                                          OK8 ())
                                    (foreach T3F23 LT3F23
                                      (setq T3F2 (car T3F23))
                                      (if (and (or OK8 (setq OK8 (> T3F2 C1INF8)))
                                               (wcmatch T3F2 PERF3))
                                          (progn
                                            (setq LT3F3 (cadr T3F23)
                                                  CC19 (strcat CC18
                                                               (substr T3F2 1 1))
                                                  C1INF9 (car (RESTA CC19 ""))
                                                  OK9 ())
                                            (foreach T3F3 LT3F3
                                              (if (and (or OK9 (setq OK9 (> T3F3
                                                                            C1INF9)))
                                                       (wcmatch T3F3 PERF3))
                                                  (write-line
                                                    (strcat (itoa (LOCL1 T1F2)) " "
                                                            (itoa (LOCL1 T1F3)) " "
                                                            (itoa (LOCL1 T2F2)) " "
                                                            (itoa (LOCL1 T2F3)) " "
                                                            (itoa (LOCL1 T3F2)) " "
                                                            (itoa (LOCL1 T3F3)))
                                                    A3)))))))))))))))))
  (close A3)
  (princ))
```

Tornem-hi, que no ha estat res! Si seguim furgant en la penúltima versió, tractant de filar més prim, ens adonarem de l'ús abusiu de **RESTA**: és admissible aprofitar aquesta funció auxiliar de **CARREGA-TRIADES** per obtenir el text **CC12** i el caràcter **C1INF2** però, anar-lo incrementant perquè l'acció de **RESTA** sobre uns textos **CC13**, **CC15**, **CC16**, **CC18** i **CC19** cada cop més llargs doni lloc a uns complements (la llista dels caràcters absents de la primera columna) cada cop més curts, és una marrada innecessària. Així que prescindirem d'aquestes variables, substituint-les per les llistes de caràcters **REST2**, **REST3**, **REST5**, **REST6**, **REST8** i **REST9**, cadascuna extreta de la precedent i amb un caràcter menys, mitjançant la funció sostractiva **REST-1**:

```
(defun REST-1 (C REST / CC)
  (foreach R (reverse REST)
    (if (/= R C) (setq CC (cons R CC))))
  CC)

(defun FES-SUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1 OK2
                      LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1 LT3F23 REST2 C1INF2
                      PERF1 T1F2 LT1F3 REST3 C1INF3 REST5 C1INF5 PERF2 T2F2 LT2F3
                      REST6 C1INF6 REST8 C1INF8 PERF3 T3F2 LT3F3 REST9 C1INF9)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  ..............................................................
```

```
(setq LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
      LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2)
      LT3 (CARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3)
      REST2 (RESTA (strcat (substr T1F1 1 1) (substr T2F1 1 1))
                           (substr T3F1 1 1))
      C1INF2 (car REST2)
      PERF1 (COMP (list T2F1 T3F1))
      A3 (open (strcat RUTA "SN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
               "w"))
(foreach T1F23 LT1F23
  (setq T1F2 (car T1F23))
  (if (and (> T1F2 C1INF2) (wcmatch T1F2 PERF1))
      (progn
        (setq LT1F3 (cadr T1F23)
              REST3 (REST-1 (substr T1F2 1 1) REST2)
              C1INF3 (car REST3))
        (foreach T1F3 LT1F3
          (if (and (> T1F3 C1INF3) (wcmatch T1F3 PERF1))
              (progn
                (setq REST5 (REST-1 (substr T1F3 1 1) REST3)
                      C1INF5 (car REST5)
                      PERF2 (COMP (list T1F1 T1F2 T1F3 T3F1)))
                (foreach T2F23 LT2F23
                  (setq T2F2 (car T2F23))
                  (if (and (> T2F2 C1INF5) (wcmatch T2F2 PERF2))
                      (progn
                        (setq LT2F3 (cadr T2F23)
                              REST6 (REST-1 (substr T2F2 1 1) REST5)
                              C1INF6 (car REST6))
                        (foreach T2F3 LT2F3
                          (if (and (> T2F3 C1INF6) (wcmatch T2F3 PERF2))
                              (progn
                                (setq REST8 (REST-1 (substr T2F3 1 1) REST6)
                                      C1INF8 (car REST8)
                                      PERF3 (COMP (list T1F1 T1F2 T1F3 T2F1 T2F2
                                                        T2F3)))
                                (foreach T3F23 LT3F23
                                  (setq T3F2 (car T3F23))
                                  (if (and (> T3F2 C1INF8) (wcmatch T3F2 PERF3))
                                      (progn
                                        (setq LT3F3 (cadr T3F23)
                                              REST9 (REST-1 (substr T3F2 1 1)
                                                            REST8)
                                              C1INF9 (car REST9))
                                        (foreach T3F3 LT3F3
                                          (if (and (= (> T3F3 C1INF9)
                                                      (wcmatch T3F3 PERF3))
                                              (write-line
                                                (strcat (itoa (LOCL1 T1F2)) " "
                                                        (itoa (LOCL1 T1F3)) " "
                                                        (itoa (LOCL1 T2F2)) " "
                                                        (itoa (LOCL1 T2F3)) " "
                                                        (itoa (LOCL1 T3F2)) " "
                                                        (itoa (LOCL1 T3F3)))
                                                A3)))))))))))))
(close A3)
(princ))
```

Ara, en els 25 minuts de sempre, ens sortien **3.650** registres: si es vol, una minsa recuperació respecte als 3.640 de la segona versió, però una davallada en relació als 3.360 de la tercera i última, i no cal insistir en què encara no havíem pogut batre el rècord discretíssim de la primera. I si provéssim d'incorporar en aquesta les variables binàries **OK**? No: ja ni havia prou de jocs amb la lletra menuda, que gairebé no tenien cap repercussió en la velocitat de processat; si volíem un canvi qualitatiu, la mare dels ous calia buscar-la en alguna decisió aparentment òbvia. I, veient que amb **wcmatch** com a eina no hi havia manera de millorar el rendiment, vam pensar a substituir la funció **CARREGA-TRIADES** per una variant **KARREGA-TRIADES** on els valors **TF1**, **TF2** i **TF3** de la llista resultant, en lloc de textos integrats

per 9 caràcters numèrics diferents, fossin llistes amb aquests caràcters numèrics, mantenint la mateixa ordenació però independents. Sobre la base de l'última versió (la més adaptable al canvi, tot i haver copsat que no era la millor), en lloc de comparar textos, utilitzant **wcmatch** amb els perfils negatius compostos per **COMP**, podríem detectar la presència de caràcters repetits en cadascuna de les 9 columnes (o el que és equivalent: detectar si algun caràcter "1", "2"... "9" no hi figura). **KARREGA-TRIADES** seria idèntica a **CARREGA-TRIADES**, tret de les línies subratllades:

```
(defun KARREGA-TRIADES (A M / TF1-1 TF1-2 TF1-3 TF2-1 TF2-2 TF2-3 TF3-1 TF3-2
                              TF3-3 LC3-1 LC3-2 LC3-3 LTF3 LTF23 LTF123)
  (setq A (open A "r") TF1 (nth M L1)
        TF1-1 (substr TF1 1 3) TF1-2 (substr TF1 4 3) TF1-3 (substr TF1 7 3)
        TF1 (STR->LC TF1))
  (while (setq TF2 (read-line A))
         (setq TF2 (nth (atoi TF2) L1) TF2-1 (substr TF2 1 3)
               TF2-2 (substr TF2 4 3) TF2-3 (substr TF2 7 3)
               TF2 (STR->LC TF2)
               LC3-1 (RESTA TF1-1 TF2-1) LTF3-1 (PERM LC3-1 (substr TF2-1 1 1)))
         (if LTF3-1
             (progn
               (setq LC3-2 (RESTA TF1-2 TF2-2) LTF3-2 (PERM LC3-2 ())
                     LC3-3 (RESTA TF1-3 TF2-3) LTF3-3 (PERM LC3-3 ())
                     LTF23 () LTF3 ())
               (foreach TF3-1 LTF3-1
                 (foreach TF3-2 LTF3-2
                   (foreach TF3-3 LTF3-3
                     (setq TF3 (strcat TF3-1 TF3-2 TF3-3)
                           TF3 (STR->LC TF3)
                           LTF3 (cons TF3 LTF3))))))
               (setq LTF23 (list TF2 (reverse LTF3))
                     LTF123 (cons LTF23 LTF123)))))
  (close A)
  (cons TF1 (reverse LTF123)))
```

Com haureu vist, l'autèntica novetat era el recurs a la nova funció **STR->LC**, que transformava els antics textos **TF1**, **TF2** i **TF3** en llistes formades pels caràcters components; la resta eren modificacions obligades per aquesta transformació, com la determinació de **TF2-2** i **TF2-3**, que abans subordinàvem a l'existència de **LTF3-1** i que ara havíem de deixar feta abans que **TF2** canviés de format, o la variable **TF3**, que abans obviàvem i que ara explicitàvem per raons estètiques, en semblar que una expressió com **(setq LTF3 (cons (STR->LC (strcat TF3-1 TF3-2 TF3-3)) LTF3))** resultava excesiva. Pel que fa a la funció **STR->LC**, aquí la teniu:

```
(defun STR->LC (STR / LC K)
  (setq K 10)
  (repeat 9 (setq K (1- K) LC (cons (substr STR K 1) LC))))
```

Amb el nou format de **TF1**, **TF2**, **TF3** i variables derivades, el procés de verificació de compatibilitat per columnes amb cada nova fila introduïda semblava més simple, perquè no es basava en un patró **wcmatch** que anava creixent sinó en un repertori de caràcters numèrics inèdits que anava minvant amb cada nova fila. Aquesta "resta" o llista de caràcters que encara no havíen aparegut es limitava a la primera columna i el seu primer membre (que, per construcció, era el menor) únicament servia per fer un garbellament previ a **wcmatch** (únic dispositiu capacitat per determinar la compatibilitat de la fila actual), per tal de saltar-nos files clarament inviables en el nostre context de solucions normalitzades) i evitar un accés innecessari al dispositiu més sofisticat. En les primeres versions obteníem la llista aprofitant la funció **RESTA** més enllà de la seva utilització a **CARREGA-TRIADES**, cosa que en la última ja només fèiem per inicialitzar la sèrie de llistes **REST...**, completada amb la més àgil funció **REST-1**, però en allò essencial seguíem igual: la barrera prèvia es basava en una llista referida a la primera columna i el nucli dur **wcmatch** obeïa a altres principis. Doncs bé, el que ara preteníem era que el filtratge previ de files òbviament inviables i el dispositiu detector d'incompatibilitats poguessin funcionar amb una única estructura de dades: ampliaríem les variables **REST...**, que passarien a ser llistes de 9 llistes on emmagatzemaríem els caràcters inèdits de cadascuna de les columnes. Inicialitzaríem la sèrie amb la funció **LCRESTA2** (com **(RESTA CC12)** però ampliada a totes les columnes) per definir **REST2**, i a cada nova fila construiríem l'actualitzada **REST...** amb **REST-1** (funció homònima de la creada

en la versió precedent i també ampliada a les nou columnes), de manera que **C1INF2** fos **(caar REST2)** i en general **C1INF...** fos **(caar REST...)**. Pel que fa al detector d'incompatibilitats, també treballaria sobre **REST...**: a diferència de **wcmatch**, amb una referència negativa cada cop més complicada, la funció **COMPARA** ho tindria cada cop més fàcil amb una referència positiva **REST...** que progressivament s'escurçava.

Tot plegat semblava augurar un gran salt endavant i que havíem assolit la versió definitiva... Però no avancem conclusions i veiem l'adaptació de **FES-SUDOKUS** al nou format de **TF1**, **TF2**, **TF3** i variables subordinades (com de costum, ometrem la part introductòria, idèntica a les versions anteriors), precedida de les funcions auxiliars citades, **LCRESTA2**, **COMPARA** i **REST-1**, i de **LC->STR** (inversa de **STR->LC**):

```
(defun LCRESTA2 (/ LLC LCR LLCR)
; (setq LLC (list (list (nth 0 T1F1) (nth 0 T2F1) (nth 0 T3F1))
;                 (list (nth 1 T1F1) (nth 1 T2F1) (nth 1 T3F1))
;                 ........................................
;                 (list (nth 8 T1F1) (nth 8 T2F1) (nth 8 T3F1)))
  (setq LLC (mapcar '(lambda (T11 T21 T31)
                       (list T11 T21 T31))
                    T1F1 T2F1 T3F1))
  (foreach LC (reverse LLC)
    (setq LCR ())
    (foreach C C9A1 ; C9A1 = '("9" "8" "7" "6" "5" "4" "3" "2" "1")
      (if (not (member C LC)) (setq LCR (cons C LCR))))
    (setq LLCR (cons LCR LLCR))))

(defun COMPARA (LC REST / OK)
  (setq OK T)
  (mapcar '(lambda (C EST)
             (if OK (setq OK (member C EST))))
          LC REST)
  OK)

(defun REST-1 (LC REST / LCP)
  (mapcar '(lambda (C EST)
             (setq LCP ())
             (foreach E (reverse EST)
               (if (/= E C) (setq LCP (cons E LCP))))
             LCP)
          LC REST))

(defun LC->STR (LC / FC)
  (setq FC "")
  (foreach C LC (setq FC (strcat FC C))))

(defun FES-SUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1 OK2
                      LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1 LT3F23 REST2 C1INF2
                      T1F2 LT1F3 REST3 C1INF3 REST5 C1INF5 T2F2 LT2F3 REST6 C1INF6
                      REST8 C1INF8 T3F2 LT3F3 REST9 C1INF9)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  ........................................
  (setq LT1 (KARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1) LT1 ()
        LT2 (KARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2) LT2 ()
        LT3 (KARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3) LT3 ()
        REST2 (LCRESTA2)
        C1INF2 (caar REST2)
        A3 (open (strcat RUTA "SN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
                 "w"))
  (foreach T1F23 LT1F23
    (setq T1F2 (car T1F23))
    (if (and (>= (car T1F2) C1INF2)
             (COMPARA T1F2 REST2))
        (progn
          (setq LT1F3 (cadr T1F23)
                REST3 (REST-1 T1F2 REST2)
                C1INF3 (caar REST3))
```

```
          (foreach T1F3 LT1F3
            (if (and (>= (car T1F3) C1INF3)
                     (COMPARA T1F3 REST3))
                (progn
                  (setq REST5 (REST-1 T1F3 REST3)
                        C1INF5 (caar REST5))
                  (foreach T2F23 LT2F23
                    (setq T2F2 (car T2F23))
                    (if (and (>= (car T2F2) C1INF5)
                             (COMPARA T2F2 REST5))
                        (progn
                          (setq LT2F3 (cadr T2F23)
                                REST6 (REST-1 T2F2 REST5)
                                C1INF6 (caar REST6))
                          (foreach T2F3 LT2F3
                            (if (and (>= (car T2F3) C1INF6)
                                     (COMPARA T2F3 REST6))
                                (progn
                                  (setq REST8 (REST-1 T2F3 REST6)
                                        C1INF8 (caar REST8))
                                  (foreach T3F23 LT3F23
                                    (setq T3F2 (car T3F23))
                                    (if (and (>= (car T3F2) C1INF8)
                                             (COMPARA T3F2 REST8))
                                        (progn
                                          (setq LT3F3 (cadr T3F23)
                                                REST9 (REST-1 T3F2 REST8)
                                                C1INF9 (caar REST9))
                                          (foreach T3F3 LT3F3
                                            (if (and (= (car T3F3) C1INF9)
                                                     (COMPARA T3F3 REST9))
                                                (write-line
                                                  (strcat
                                                    (itoa (LOCL1 (LC->STR T1F2)))
                                                  " " (itoa (LOCL1 (LC->STR T1F3)))
                                                  " " (itoa (LOCL1 (LC->STR T2F2)))
                                                  " " (itoa (LOCL1 (LC->STR T2F3)))
                                                  " " (itoa (LOCL1 (LC->STR T3F2)))
                                                  " " (itoa (LOCL1 (LC->STR T3F3))))
                                                  A3)))))))))))))))))))
  (close A3)
  (princ))
```

Després de tants i tant il·lusionats preparatius, aquest **FES-SUDOKUS** se'ns penjava abans d'obrir-se l'arxiu **A3** (en el segon ús d'aquesta variable), en quedar ofegada perquè la memòria disponible resultava insuficient per emmagatzemar simultàniament llistes tan llargues com **L1**, **LT1F23**, **LT2F23** i **LT3F23**. Ja no citem **LT1**, **LT2** ni **LT3** perquè, després del primer fracàs, la primera idea que ens va venir al cap va ser desempallegar-nos-en de les tres tan aviat com poguéssim (en el codi que us hem ofert ja hi figura aquesta supressió), però únicament amb això no n'hi havia prou. D'altra banda, no podíem prescindir de **L1**, perquè per escriure en **A3** necessitàvem aquesta llista: recordeu que a la funció **LOCL1** hi tenim l'expressió **(member E L1)**.

Però la pressumpció que el nou enfocament havia de comportar un guany de velocitat considerable ens va fer abandonar l'obsessió per la grandària de l'arxiu: sent més precisos, per la longitud d'uns registres que, si representaven les posicions de les files a **L1**, es xifrava en uns 40 caràcters de mitjana, mentre que representant el seu contingut se n'anàven a **6 × 9 = 54** caràcters (i això tenint present que les 6 files podien representar-se sense solució de continuïtat, en tenir una longitud constant de 9 caràcters a diferència de les posicions, la longitud variable de les quals obligava a inserir 5 espais separadors). Transferint directament a l'arxiu el contingut de les files 2ª, 3ª, 5ª, 6ª, 8ª i 9ª, podríem prescindir de la funció **LOCL1** i, de retruc, de la llista **L1**, que podríem anul·lar just després de l'accés **(KARREGA-TRIADES A3 M3)**. D'aquesta manera, el final de **FES-SUDOKUS** quedaria així:

```
                                    .............................
                                    (foreach T3F3 LT3F3
                                      (if (and (> T3F3 C1INF9)
                                               (COMPARA T3F3 REST9))
```

```
                                                    (write-line
                                                      (strcat (LC->STR T1F2)
                                                              (LC->STR T1F3)
                                                              (LC->STR T2F2)
                                                              (LC->STR T2F3)
                                                              (LC->STR T3F2)
                                                              (LC->STR T3F3))
                                                      A3)))))))))))))))))
  (close A3)
  (princ))
```

Però no havia servit de res desfer-nos de **LT1**, **LT2**, **L1** i **LT3**: seguia faltant espai de memòria per emmagatzemar **LT1F23**, **LT2F23** i **LT3F23**. Resultava evident que alguna diferència d'ocupació havia d'haver entre un *string* de 9 caràcters (per exemple, el primer membre de **L1**: **"123456789"**), que era la informació amb què representàvem una fila en les primeres versions de **FES-SUDOKUS**, i una llista constituïda per 9 *strings* d'un caràcter (per exemple, **'("1" "2" "3" "4" "5" "6" "7" "8" "9")**), que era la que adoptàvem ara. Ens preguntàvem si, substituint-la per llistes formades per 9 enters (per exemple, **'(1 2 3 4 5 6 7 8 9)**), podríem reduir-la fins al punt de desblocar la situació. I per ser gradualistes, apurant totes les possibilitats, vam decidir de reprendre la penúltima versió pel que fa a la no supressió de **L1** i al retorn a un arxiu de posicions. Ens limitarem a presentar les dues funcions que ens van permetre el canvi, evitant presentar un altra vegada el codi de **KARREGA-TRIADES** i de **FES-SUDOKUS**, perquè en la primera funció només cal substituir **STR->LC** per **STR->LN** i, en la segona, **LC->STR** per **LN->STR** i **LCRESTA2** per **LNRESTA2**. Pel que fa a les dues primeres, veureu que les úniques novetats són la presència d'**atoi** a la primera i d'**itoa** a la segona, i en la tercera ens limitem a substuir la llista **C9A1** (de valor **'("9" "8" "7" "6" "5" "4" "3" "2" "1")**) per **'(9 8 7 6 5 4 3 2 1)**. Per tal que pugueu localitzar més ràpidament aquestes petites modificacions, les hem subratllades:

```
(defun STR->LN (STR / LN K)
  (setq K 10)
  (repeat 9 (setq K (1- K) LN (cons (atoi (substr STR K 1)) LN))))

(defun LN->STR (LN / FC)
  (setq FC "")
  (foreach N LN (setq FC (strcat FC (itoa N)))))

(defun LNRESTA2 (/ LLN LNR LLNR)
; (setq LLN (list (list (nth 0 T1F1) (nth 0 T2F1) (nth 0 T3F1))
;                 (list (nth 1 T1F1) (nth 1 T2F1) (nth 1 T3F1))
;                 ..........................................
;                 (list (nth 8 T1F1) (nth 8 T2F1) (nth 8 T3F1)))
  (setq LLN (mapcar '(lambda (T11 T21 T31)
                       (list T11 T21 T31))
                    T1F1 T2F1 T3F1))
  (foreach LN (reverse LLN)
    (setq LNR ())
    (foreach N '(9 8 7 6 5 4 3 2 1)
      (if (not (member N LN)) (setq LNR (cons N LNR))))
    (setq LLNR (cons LNR LLNR))))
```

Tanmateix, per reconèixer aquesta versió amb un simple cop d'ull, diferenciant-la de les dues precedents i de les que seguiran, reproduirem el final de **FES-SUDOKUS**:

```
                                                    ..............................
                                                    (write-line
                                                      (strcat
                                                        (itoa (LOCL1 (LN->STR T1F2)))
                                                      " " (itoa (LOCL1 (LN->STR T1F3)))
                                                      " " (itoa (LOCL1 (LN->STR T2F2)))
                                                      " " (itoa (LOCL1 (LN->STR T2F3)))
                                                      " " (itoa (LOCL1 (LN->STR T3F2)))
                                                      " " (itoa (LOCL1 (LN->STR T3F3))))
                                                        A3)))))))))))))))))
  (close A3)
  (princ))
```

L'havíem encertada en suposar que les llistes de caràcters (en realitat hauríem de parlar de textos o *strings* constituïts per un únic caràcter) consumien força més memòria que les llistes d'enters (almenys d'enters de rang 1... 9), perquè ara, si més no, l'execució de **FES-SUDOKUS** proseguia. Però si recordem que el canvi de rumb havia estat dictat per la voluntat de construir més de pressa l'arxiu de solucions normalitzades, aquesta victòria era pírrica, perquè el ritme encara era més lent que en les primeres versions. ¿Seria per no haver gosat anul·lar **L1**, cosa que ens obligava a tornar a un arxiu amb el contingut de les files? Podia ser, però volíem explorar les alternatives de forma gradual: de primer, mantindríem **L1** durant tota l'execució, limitant-nos a canviar les expressions finals de **FES-SUDOKUS** que donen format a l'arxiu de solucions normalitzades, per veure si aquest simple canvi ja comportava un avenç; després, ja veuríem si suprimint **L1** l'execució experimentava guanys addicionals. Així que, de moment, sols vam canviar el final de **FES-SUDOKUS**:

```
                                    ..............................
                                    (foreach T3F3 LT3F3
                                      (if (and (> T3F3 C1INF9)
                                               (COMPARA T3F3 REST9))
                                          (write-line
                                            (strcat (LN->STR T1F2)
                                                    (LN->STR T1F3)
                                                    (LN->STR T2F2)
                                                    (LN->STR T2F3)
                                                    (LN->STR T3F2)
                                                    (LN->STR T3F3))
                                             A3)))))))))))))))))
  (close A3)
  (princ))
```

Sembla que anàvem pel bon camí, perquè el ritme de construcció de l'arxiu superava àmpliament el de les primeres versions. Paradoxalment, quan volíem batre el rècord tot just assolit, alliberant memòria amb l'anul·lació de la llista **L1**, no només no en va resultar cap guany de velocitat sinó una lleugera regressió, tot i mantenir l'indubtable avantatge sobre les primeres versions. Els migradíssims coneixements d'informàtica de l'autor no li permeten justificar aquest fenòmen: ¿podria ser que l'operació consumís un temps que, no existint ja cap situació d'ofec, no arribés a influir en la velocitat de processat?. Deixem aquesta disquisició per a qui pugui respondre-la amb més autoritat, i reprenem el discurs. L'autor, que mai ha tingut el més mínim pudor d'amagar la seva ignorància en moltes matèries en què tanmateix li ha vingut de gust fer de tastaolletes, però que tampoc no acostuma a pecar de falsa modèstia a l'hora d'exhibir perseverància i paciència com a trets destacats de la seva personalitat, de seguida es va adonar que alguna cosa grinyolava, i que s'ensumava haver fet una marrada èpica per acabar tornant a un punt molt proper al de partença (més endavant veureu que aquests freqüents circumloquis heurístics han posat malnom a algun capítol com el 5, postil·lat "roda el món i torna al Born").

En definitiva, s'adonava que la pretesa superioritat del dispositiu **COMPARA** (basat en **member**) sobre **wcmatch** només era una pressumpció no ratificada per l'experiència (pitjor: una pressumpció falsa desmentida indirectament per l'experiència) perquè, en fer-lo treballar en les mateixes condicions que **wcmatch** (és a dir, passant els resultats a un arxiu que recollia les posicions de les files en **L1**), revelava ser més lent. Si alguna virtut havia tingut atomitzar el contingut de les files de les tríades compatibles, a efectes de comparació per columnes, havia estat animar-nos a transgredir el tabú dels arxius de solucions transcrivint sense més el contingut de les files 2, 3, 5, 6, 8 i 9, de primer (amb llistes de caràcters) per veure si, podent prescindir de **L1**, aconseguíem desblocar l'execució, i després (amb llistes d'enters) per veure si, passant de registrar posicions a registrar continguts, la construcció de l'arxiu anava més ràpida: potser sense abordar el canvi de textos (comparats amb **wcmatch**) per llistes (comparades amb **member**) ni se'ns hagués acudit que el format de l'arxiu (que implicava passar o no pel tràmit **(itoa (LOCL1 ...))**) podia influir més en la velocitat del procés que el mètode de comparació adoptat. En qualsevol cas i malgrat l'evidència indirecta, no podíem eludir la comprovació experimental, ni que fos per quantificar el guany, així que vam recuperar l'última versió que treballava amb files-text, deixant el primer argument de **write-line** tal com podeu veure en aquest nou final de **FES-SUDOKUS**, reduït a la mínima expressió:

```
                                    ..................................
                                    (foreach T3F3 LT3F3
                                      (if (and (> T3F3 C1INF9)
                                               (wcmatch T3F3 PERF3))
```

```
                                        (write-line (strcat T1F2 T1F3
                                                            T2F2 T2F3
                                                            T3F2 T3F3)
                                                    A3)))))))))))))))))
  (close A3)
  (princ))
```

Els resultats ens van deixar bocabadats: la simple substitució del primer argument de **write-line** havia multiplicat per **69** la velocitat de construcció de l'arxiu de solucions normalitzades; és a dir que en un mateix període de temps el nombre de registres creats s'havia multiplicat per **69**, tot i que els registres reproduint el contingut de 6 files, sense solució de continuïtat, eren un 35% més llargs que els integrats per les 6 posicions a **L1**, separades per espais (de fet, l'espai ocupat per l'arxiu no s'havia multiplicat per 67 sinó per 88). Com comprendreu, la causa de l'avenç no rau tant en el valor d'aquest argument com en la feina d'avaluar els arguments de **strcat** (els textos que ha de concatenar), ara servits per variables i abans per complicades expressions. Davant d'aquest fet tan evident (*a posteriori*, tot resulta evident), els canvis precedents tenien una repercussió insignificant: les petites diferències entre l'última versió de **FES-SUDOKUS** (la que ens ha servit de conill d'Índies i, de retruc, per treure'n l'entrellat) i la primera, de només 30 registres sobre 3.680 (0,8%) en 25 minuts, s'inverteixen quan ambdues versions adopten l'expressió **(write-line (strcat T1F2 T1F3 T2F2 T2F3 T3F2 T3F3) A3)**, perquè en 25 minuts la primera versió es queda en 251.200 registres i l'última en 253.300 (la diferència segueix representant el 0,8% però ha canviat de signe), a frec dels 255.091 registres que componen l'arxiu (com no trigarem a veure). Si ens haguéssim de plantar aquí, potser aquesta petita diferència justificaria que seguíssim amb l'última versió, però veurem com encara és possible reblar el clau, aconseguint de completar l'arxiu en 25 minuts. I com que curiosament ambdues versions coincidiran en aquests 25 minuts, restant transcendència a les oscil·lacions que hem observat, fet i fet ens quedarem amb la primera, força més simple. No us estranyi, doncs, si d'ara endavant les variables **REST2**, **C1INF2**, **REST3**, **C1INF3**, **REST5**, **C1INF5**, **REST6**, **C1INF6**, **REST8**, **C1INF8**, **REST9** i **C1INF9** són absents del codi: ja no les necessitem.

L'èxit assolit (éxit relatiu, ja que inventariar totes les solucions normalitzades segueix quedant fora del nostre abast) esdevindria més satisfactori si no fos per la major ocupació de l'arxiu de continguts, tema que convindria no tancar encara, estudiant les possibilitats d'aproximar aquesta ocupació a la del format obsolet. Però abans d'intentar el nou repte seria bo consolidar la victòria, abordant una qüestió transcendental: el disseny d'una rutina que permetés a l'usuari d'accedir còmodament a les solucions inventariades, perquè no només les files comuns 1, 4 i 7 segueixen codificades (per no fer més llarg el nom de l'arxiu, en el qual estan integrades) sinó que la lectura directa de l'arxiu seria feixuga, en no tractar-se de files seguides i no haver-hi espais separadors; a més, encara que no fos així, no seria acceptable que l'usuari no tingués accés a una presentació amb l'escaquer 9×9 mostrant l'alineació per files i columnes, i la concentració en requadres 3×3.

A la rutina l'hem anomenada **VIS-XSN**, i en la sol·licitud del tipus d'arxiu que ha d'obrir l'usuari parlarem de "arxiu eXplícit de Solucions Normalitzades". Per què? Doncs perquè encara no havíem explicat que, després que haguéssim decidit anomenar SN-<pF1>-<pF4>-<pF7> (<pF1>, <pF4> i <pF7> representen les posicions a **L1** de les files comuns 1, 4 i 7) els arxius de solucions normalitzades en "format posició", les provatures en "format contingut" les vam voler diferenciar anomenant l'arxiu XSN-<pF1>-<pF4>-<pF7>, per no confondre ni destruir mútuament els arxius d'ambdós formats referits a les mateixes col·leccions de tríades compatibles KOMPAT-<pF1>, KOMPAT-<pF2> i KOMPAT-<pF3>, anàlogament al canvi de notació entre aquests darrers i el format obsolet COMPAT-<pF1>, COMPAT-<pF2> i COMPAT-<pF3>. L'hem dissenyada de manera que l'usuari pugui optar entre veure solucions concretes (referenciades per mitjà de la seva posició a l'arxiu, raó per la qual la primera cosa que fa **VIS-XSN** és recórrer tots els registres per poder-lo informar del nombre de solucions, que serà elevat encara que no haguem pogut esperar a que **FES-SUDOKUS** completés la seva feina i haguem cancel·lat el procés forçant el tancament d'AutoCAD amb l'ajut de l'Administrador de Tasques) o anar-les veient seqüencialment, agrupades de tres en tres. Sobre les funcions auxiliars, ens limitarem a dir que **PREFORM** fragmenta els registres llegits en trossos de 9 caràcters, corresponents a les files 2, 3, 5, 6, 8 i 9; **FORMAT** espaia els caràcters a fi de millorar-ne la visualització, separant encara més els tercets horitzontals, i **IMPR** escriu en pantalla (en ordre invers) les línies que representen files (de tres solucions consecutives o d'una de sola), separant més les corresponents als diferents tercets verticals. Veieu-ne el codi:

```
(defun PREFORM ()
  (setq F2 (substr F 1 9)  F3 (substr F 10 9)
        F5 (substr F 19 9) F6 (substr F 28 9)
        F8 (substr F 37 9) F9 (substr F 46 9)))

(defun FORMAT (L / F FT)
  (foreach QF L
    (setq F (eval QF) FT "   " J 0)
    (repeat 3
      (setq FT (strcat FT " "))
      (repeat 3 (setq J (1+ J) FT (strcat FT " " (substr F J 1)))))
    (set QF FT)))

(defun IMPR (F1 F2 F3 F4 F5 F6 F7 F8 F9)
  (terpri)
  (terpri) (princ F9) (terpri) (princ F8) (terpri) (princ F7) (terpri)
  (terpri) (princ F6) (terpri) (princ F5) (terpri) (princ F4) (terpri)
  (terpri) (princ F3) (terpri) (princ F2) (terpri) (princ F1) (terpri))

(defun VIS-XSN (/ L1 RUTA A A1 A1P F I J K O F1 F2 F3 F4 F5 F6 F7 F8 F9
                  F31 F32 F33 F34 F35 F36 F37 F38 F39)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)))
  (while (progn
           (setq A (getstring (strcat "\n.\nEntreu el nom d'algun arxiu eXplícit "
                                      "de Solucions Normalitzades\n(XSN-pF1-pF4-"
                                      "pF7), sense l'extensió .txt: XSN-")))
           (not (setq A1 (findfile (strcat RUTA "XSN-" A ".txt")))))
         (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no s'hi ha "
                         "trobat cap arxiu \nXSN-" A ".txt. REPETIU!")))
  (setq A1P (open A1 "r")
        F A F1 (itoa (atoi F))
        F (substr F (+ (strlen F1) 2)) F4 (itoa (atoi F))
        F (substr F (+ (strlen F4) 2)) F7 (atoi F)
        F1 (nth (atoi F1) L1) F4 (nth (atoi F4) L1) F7 (nth F7 L1) K 0)
  (FORMAT (list 'F1 'F4 'F7))
  (while (read-line A1P) (setq K (1+ K)))
  (close A1P)
  (terpri)
  (prompt (strcat "\nL'arxiu XSN-" A ".txt té " (itoa K)
                  " solucions normalitzades:"))
  (prompt "\n1) Podeu veure-les totes seqüencialment, de 3 en 3.")
  (prompt (strcat "\n2) Podeu veure-les reclamant-les per la seva posició "
                  "(1 a " (itoa K) ").\n"))
  (while (progn
           (setq O (getint "Opció 1 o 2: "))
           (or (< O 1) (> O 2)))
         (prompt "\n  REPETIU! "))
  (if (= O 1)
      (progn
        (setq A1P (open A1 "r") I 3)
        (while (and (= I 3) (> (setq I (if (> (+ O 2) K) (rem K 3) 3)) 0))
               (prompt "\n                    Per seguir, polseu qualsevol tecla ")
               (prompt "(<Esc> per acabar)")
               (grread)
               (prompt (if (> I 1)
                           (strcat "\nSolucions " (itoa O) " "
                             (if (> I 2) "a" "i") " " (itoa (+ O (1- I))) ":")
                           (strcat "\n Solució " (itoa O) ":")))
               (setq F31 "" F32 "" F33 "" F34 "" F35 "" F36 "" F37 "" F38 ""
                     F39 "")
               (repeat I
                 (setq O (1+ O) F (read-line A1P))
                 (PREFORM)
                 (FORMAT (list 'F2 'F3 'F5 'F6 'F8 'F9))
```

```
                (setq F31 (strcat F31 F1) F32 (strcat F32 F2) F33 (strcat F33 F3)
                      F34 (strcat F34 F4) F35 (strcat F35 F5) F36 (strcat F36 F6)
                      F37 (strcat F37 F7) F38 (strcat F38 F8)
                      F39 (strcat F39 F9)))
              (IMPR F31 F32 F33 F34 F35 F36 F37 F38 F39))
        (close A1P))
      (while F
             (setq A1P (open A1 "r"))
             (terpri)
             (while (progn
                      (setq F (getreal (strcat "Entreu núm. solució, entre 1 i "
                                               (itoa K) ", "
                                               "(<CR> o <blanc> per acabar): ")))
                      (and F (or (< F 1) (> F K))))
                    (prompt "\n  REPETIU! "))
             (if F (progn
                     (setq A1P (open A1 "r")
                           O (fix F) F (repeat O (read-line A1P)))
                     (PREFORM)
                     (FORMAT (list 'F2 'F3 'F5 'F6 'F8 'F9))
                     (IMPR F1 F2 F3 F4 F5 F6 F7 F8 F9)
                     (close A1P)))))
  (princ))
```

Ara, que ja tenim una base acceptable, és quan podem abordar el tema de l'ocupació dels arxius eXplícits de Solucions Normalitzades, d'indiscutible superioritat pel que fa a rapidesa de construcció però que arrosseguen el pecat original de ser més voluminosos que els primitius arxius de posicions. No és que en termes relatius un 35% més sigui excessiu però, considerant les dimensions que poden assolir aquests arxius (per bé que sempre els podem transferir a un dispositiu extern de memòria, un cop construïts del tot o parcialment en disc) qualsevol estalvi serà benvingut. L'única precaució a tenir en compte, en la cerca de possibilitats de compactació, és que el procés de descompactació recíproc no representi un alentiment exagerat: que l'aparició de solucions en pantalla no suposi llargues esperes per a l'usuari. L'objectiu que ens hem marcat tampoc no és res de l'altre món, perquè a propòsit de les nou files que facilita cada registre (sis d'explícites i tres d'implícites en el nom de l'arxiu) tenim una informació valuosa: defineixen una SUDOKU-SOLUCIÓ. Això vol dir que els 81 caràcters numérics no són arbitraris sinó que obeeixen a regles molt precises, raó per la qual tenen un alt nivell de redundància que podem rebaixar, suprimint-ne alguns. Com ja haurà endevinat el lector, podríem eliminar una fila i una columna, això sense tenir en compte els condicionaments específics dels requadres 3×3, i per tal d'adaptar-nos millor a l'estructura de l'exploració que es realitza a **FES-SUDOKUS**, prescindirem de la 9ª fila i de la 9ª columna, peró no abordarem les dues reduccions de forma simultània, sinó succesiva: així podrem controlar millor l'impacte dels canvis sobre la funció esmentada i sobre **VIS-XSN**.

En eliminar la fila 9 dels registres de l'arxiu de solucions (passant de **54** a **45** caràcters), augmentarem encara un pèl més la velocitat d'actuació de **FES-SUDOKUS**, però no només per prescindir del sisè fragment **T3F3** en l'encadenament del primer argument de la funció terminal **write-line**, sinó perquè, tan bon punt trobem una fila **T3F3** compatible amb les 8 precedents ja no caldrà seguir buscant-ne d'altres: de fet, aquesta circumstància l'haguéssim pogut considerar des del principi, però se'ns havia passat per alt; ha estat la reflexió sobre la possibilitat teòrica de prescindir de la 9ª fila (amb això de "teòrica" volem significar que no sempre 8 files compatibles en determinen una 9ª que també ho sigui, perquè pot passar que en una 9ª fila composta pels caràcters "1" ... "9" absents de cada columna de vuit hi hagi repeticions) que ha fet adonar-nos-en. Abans de mostrar-vos els canvis a **FES-SUDOKUS** (són tan nimis que hem cregut convenient subratllar-los), comentarem que l'experiència negativa d'abans en relació a les millores que esperàvem obtenir de la substitució de textos per files de caràcters o d'enters, ens ha inclinat a tirar pel dret i mantenir l'exploració de novenes files **T3F3** de la llista **LT3F3** fins a trobar la compatible (si existeix), introduint únicament el senyal binari **OK** per deturar-la, en lloc de compondre directament una 9ª fila "residual" i mirar si pertanyia a **LT3F3**, mètode potser més elegant. Com sempre, en veureu el final:

```
                                    .....................................
                                    (if (wcmatch T3F2 PERF3)
                                        (progn
                                          (setq LT3F3 (cadr T3F23) OK T)
```

```
                                        (foreach T3F3 LT3F3
                                          (if (and OK (wcmatch T3F3 PERF3))
                                              (progn
                                                (setq OK ())
                                                (write-line
                                                  (strcat T1F2 T1F3 T2F2 T2F3
                                                          T3F2 _)
                                                  A3))))))))))))))))))
  (close A3)
  (princ))
```

Pel que fa a **VIS-XSN**, la afectació ha estat mínima: únicament ha calgut modificar la funció auxiliar **PREFORM**, per deduir la 9ª fila **F9** de les altres vuit, i això ha comportat utilitzar a l'inici unes variables-vicari **F1***, **F4*** i **F7*** que permetessin preservar **F1**, **F4** i **F7**. Com que la resta queda igual, sols reproduirem aquest tros:

```
(defun PREFORM (/ OK L)
  (setq F2 (substr F 1 9)  F3 (substr F 10 9)
        F5 (substr F 19 9) F6 (substr F 28 9)
        F8 (substr F 37 9) F9 "" J 0)
  (repeat 9
    (setq J (1+ J) OK T L ())
    (foreach F (list F1* F2 F3 F4* F5 F6 F7* F8) (setq L (cons (substr F J 1) L)))
    (foreach C '("1" "2" "3" "4" "5" "6" "7" "8" "9")
      (if (and OK (not (member C L))) (setq OK () F9 (strcat F9 C))))))

(defun VIS-XSN (/ L1 RUTA A A1 A1P F I J K O F1 F2 F3 F4 F5 F6 F7 F8 F9
                  F1* F4* F7* F31 F32 F33 F34 F35 F36 F37 F38 F39)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)))
  (while (progn
           (setq A (getstring (strcat "\n.\nEntreu el nom d'algun arxiu eXplícit "
                                      "de Solucions Normalitzades\n(XSN-pF1-pF4-"
                                      "pF7), sense l'extensió .txt: XSN-")))
           (not (setq A1 (findfile (strcat RUTA "XSN-" A ".txt")))))
         (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no s'hi ha "
                         "trobat cap arxiu \nXSN-" A ".txt. REPETIU!")))
  (setq A1P (open A1 "r")
        F A F1 (itoa (atoi F))
        F (substr F (+ (strlen F1) 2)) F4 (itoa (atoi F))
        F (substr F (+ (strlen F4) 2)) F7 (atoi F)
        F1 (nth (atoi F1) L1) F4 (nth (atoi F4) L1) F7 (nth F7 L1)
        F1* F1 F4* F4 F7* F7 K 0)
  (FORMAT (list 'F1 'F4 'F7))
  ..............................................................
```

Com que la presentació de les solucions no ha experimentat cap retard apreciable per causa de la primera intervenció reductora en els arxius, abordarem la segona: passar de **45** a **40** caràcters per registre, encadenant 5 files de només 8 caràcters, retallada amb la qual aconseguim una procustiana igualació amb el primitiu format de posicions a **L1**. En relació a **FES-SUDOKUS**, no estàvem segurs de quina seria la millor manera de minimitzar l'impacte de l'aplicació de **(substr ... 1 8)** a **T1F2**, **T1F3**, **T2F2**, **T2F3** i **T3F2**, perquè dubtàvem entre afegir les variables locals **T1F2***, **T1F3***, **T2F2***, **T2F3*** i **T3F2***, i deixar la segona meitat de la funció com segueix,

```
  ..............................
  (foreach T1F23 LT1F23
    (setq T1F2 (car T1F23)
          T1F2* (substr T1F2 1 8))
    (if (wcmatch T1F2 PERF1)
        (progn
          (setq LT1F3 (cadr T1F23))
          (foreach T1F3 LT1F3
            (setq T1F3* (substr T1F3 1 8))
            (if (wcmatch T1F3 PERF1)
                (progn
                  (setq PERF2 (COMP (list T1F1 T1F2 T1F3 T3F1)))
```

```
                    (foreach T2F23 LT2F23
                      (setq T2F2 (car T2F23)
                            T2F2* (substr T2F2 1 8))
                      (if (wcmatch T2F2 PERF2)
                          (progn
                            (setq LT2F3 (cadr T2F23))
                            (foreach T2F3 LT2F3
                              (setq T2F3* (substr T2F3 1 8))
                              (if (wcmatch T2F3 PERF2)
                                  (progn
                                    (setq PERF3 (COMP (list T1F1 T1F2 T1F3 T2F1 T2F2
                                                            T2F3)))
                                    (foreach T3F23 LT3F23
                                      (setq T3F2 (car T3F23)
                                            T3F2* (substr T3F2 1 8))
                                      (if (wcmatch T3F2 PERF3)
                                          (progn
                                            (setq LT3F3 (cadr T3F23) OK T)
                                            (foreach T3F3 LT3F3
                                              (if (and OK (wcmatch T3F3 PERF3))
                                                  (progn
                                                    (setq OK ())
                                                    (write-line
                                                      (strcat T1F2* T1F3* T2F2*
                                                              T2F3* T3F2*)
                                                      A3)))))))))))))
  (close A3)
  (princ))
```

sobrecarregant amb l'expressió **(substr ... 1 8)** fins i tot les files que no surten a l'arxiu per no ser compatibles amb cap successora, però fent-ho només un cop, o deixar-ho tot tal com estava, afectant únicament el primer argument de **write-line**,

```
                                    .................................
                                    (foreach T3F3 LT3F3
                                      (if (and OK (wcmatch T3F3 PERF3))
                                          (progn
                                            (setq OK ())
                                            (write-line
                                              (strcat (substr T1F2 1 8)
                                                      (substr T1F3 1 8)
                                                      (substr T2F2 1 8)
                                                      (substr T2F3 1 8)
                                                      (substr T3F2 1 8))
                                              A3)))))))))))))))))))
  (close A3)
  (princ))
```

actuació que només aplica **(substr ... 1 8)** a les files que van a parar a l'arxiu, però que, si són compatibles amb diverses combinacions de successores, ho fa totes les vegades. Finalment s'ha optat pel segon procediment, atès que en els dos casos que tot seguit comentarem la seva rapidesa superava la del primer en un 45%, tot i reconèixer que dos exemples (tan pròxims, malgrat la disparitat en grandària dels arxius de solucions en què es materialitzen) són poca base per treure conclusions. El primer té una particularitat: a una de les 4.635.350 solucions normalitzades a què dóna lloc l'anomenem SOLUCIÓ CANÒNICA a partir del següent capítol, representa una de les tres opcions que oferirà el programa per dotar-se d'una SUDOKU-SOLUCIÓ a partir de la qual obtenir SUDOKUs-PROBLEMA (les altres dues són SOLUCIÓ COPIADA i SOLUCIÖ CREADA) i, si considerem la primera fila de **L1** una seqüència circular en què després del **9** vé l'**1**, totes les files responen a aquesta seqüència. La 1ª fila és **"123456789"** (posició 0 a **L1**), la 4ª és **"234567891"** (posició 46.233) i la 7ª és **"345678912"** (posició 92.464), i el corresponent arxiu XSN-0-46233-92464.txt ocupa 190.122 KBytes, té **4.635.350** registres i triga 7 hores i 50 minuts a completar-se. El segon és l'exemple de referència de totes les provatures realitzades fins ara: resulta de substituir per **"198234567"** (posició 39.600 a **L1**) la 1ª fila **"123456789"** de l'anterior, però la 4ª i la 7ª són iguals. Tanmateix, les característiques de l'arxiu XSN-39600-46233-92464 han variat completament: només ocupa 10.463 KBytes, té **255.091** registres-solució i triga 25 minuts a completar-se; si recordeu que en la primera versió de **FES-SUDOKUS** sols obteníem 3.680 registres en el mateix temps, això vol dir que la velocitat de construcció de registres s'ha multiplicat per **69**.

Però, ¿que no havíem dit ja el mateix, cinc fulls enrera, i tanmateix a l'arxiu li calia una mica més de temps?: ben cert, però en aquell moment acabàvem de donar el salt al format de continguts, resultant-ne uns registres de 54 caràcters; en canvi ara hem aconseguit comprimir-los fins a 40 caràcters, tornant a l'ocupació mitjana dels arxius de posicions, és a dir que podem parlar d'un rendiment net del **6.900%**.

I, pel que fa a la lectura de l'arxiu amb aquest format més econòmic i presentació en pantalla de les solucions, **VIS-XSN** es queda igual, modificant-se només **PREFORM**, que mostrem acompanyada de la funció auxiliar **COMPLETA**:

```
(defun COMPLETA (F)
  (setq J 0 OK T L ())
  (repeat 8 (setq J (1+ J) L (cons (substr F J 1) L)))
  (foreach C 9C
    (if (and OK (not (member C L))
        (setq OK () L (strcat F C))))
  L)

(defun PREFORM (/ OK L 9C)
  (setq 9C '("1" "2" "3" "4" "5" "6" "7" "8" "9")
        F2 (substr F 1 8) F2 (COMPLETA F2)
        F3 (substr F 9 8) F3 (COMPLETA F3)
        F5 (substr F 17 8) F5 (COMPLETA F5)
        F6 (substr F 25 8) F6 (COMPLETA F6)
        F8 (substr F 33 8) F8 (COMPLETA F8)
        F9 "" J 0)
  (repeat 9
    (setq J (1+ J) OK T L ())
    (foreach F (list F1* F2 F3 F4* F5 F6 F7* F8)
      (setq L (cons (substr F J 1) L)))
    (foreach C 9C
      (if (and OK (not (member C L))) (setq OK () F9 (strcat F9 C))))))
```

Igual que en l'anterior compactació de l'arxiu, tampoc amb aquesta la presentació de les solucions no ha experimentat cap retard sensible. El que sí podem arribar a apreciar, llegint arxius de gran longitud, és una petita pausa a l'inici o (usant l'opció d'accés sota demanda) si volem veure alguna de les últimes solucions, però això no és pas degut al format dels registres (de fet, ja passava en les versions precedents de **VIS-XSN**) sinó al fet que la funció hagi de recórrer l'arxiu de cap a peus. Per fer-vos una idea, amb l'atrotinat equip utilitzat per l'autor la pausa és de 12 segons amb l'arxiu XSN-0-46233-92464, però amb XSN-39600-46233-92464 ja és del tot imperceptible.

Abans d'acabar el capítol amb un sumari o guia pràctica d'actuació, i d'una manera similar a l'ultima rutina **FES-COMPAT-M-N-O**, que permetia de crear sèries d'arxius de tríades compatibles COMPAT-..., en vista de l'estalvi de temps i d'ocupació que hem assolit en passar al format KOMPAT-..., pensem que fóra interessant donar una alternativa a l'ús manual i repetit de l'última versió de **FES-SUDOKUS**, permetent que l'usuari amb discs de gran capacitat i possibilitats de deixar treballant el seu ordinador dies sencers posés en marxa un procés de fabricació seriada d'arxius XSN-... Podríem fixar les condicions de partença (el primer arxiu que s'hauria de crear, determinat per la posició a **L1** de les files comuns 1ª, 4ª i 7ª) i la rutina aniria generant arxius mitjançant la progressió entre candidates a 1ª fila (des de l'explicitada fins a la corresponent a la posició 40.319 de **L1**); per a cadascuna d'aquestes, la progressió entre candidates a 2ª fila (des de l'explicitada, que hauria d'ocupar a **L1** una posició superior a 40.319, fins a la corresponent a la posició 161.279, sempre que fossin compatibles amb la 1ª fila) i, per a cadascuna d'aquestes, la progressió entre candidates a 3ª fila (des de l'explicitada, que hauria d'ocupar a **L1** una posició superior a 80.639, 120.959 o 161.279, segons que la 2ª comencés per "2", "3" o "4", fins a la corresponent a la posició 282.239, sempre que fossin compatibles amb la 1ª i 2ª files). Fins i tot, perquè l'usuari no hagués d'estar pendent dels arxius existents, podríem decidir que en aquesta dinàmica només es fabriquessin els arxius XSN-... inexistents, donant per bons els que ja figuressin a la ruta de cerca d'AutoCAD. I encara podríem fer més: que, en lloc de l'actuació sistemàtica que estava implícita en allò que hem fet fins ara (per crear els arxius XSN-... calia disposar dels arxius KOMPAT-... implicats), la construcció d'un l'arxiu XSN-<pF1>-<pF4>-<pF7> (conjunt de solucions normalitzades en què la 1ª fila fos <pF1>, la 4ª fos <pF4> i la 7ª fos <pF7>) no depengués de la

prèvia existència dels arxius KOMPAT-<pF1>, KOMPAT-<pF4> i KOMPAT-<pF7> (conjunts de tríades compatibles que comencen amb les files <pF1>, <pF4> i <pF7>), sinó que l'absència d'aquests ingredients pogués ser resolta sobre la marxa, igual que hem proposat per als arxius XSN-... (respectar els existents i crear els que falten). Com que aquí sí que hauríem de treballar per etapes més o menys llargues i l'única manera d'acabar cada sessió seria tancar AutoCAD amb l'ajut de l'Administrador de Tasques, deixant incomplet l'últim arxiu XSN-..., la norma obligada seria esborrar aquest arxiu, tret que no pretenguéssim ser exhaustius i només volguéssim disposar d'unes quantes solucions normalitzades. Pel que fa a l'últim arxiu KOMPAT-..., tot dependrà de quin dels dos arxius sigui el més recent, observant-ne el llistat en la finestra de Windows, una vegada cancel·lada l'execució: si l'últim KOMPAT-... és anterior a l'últim XSN-..., llavors el podem deixar perquè s'haurà completat; si és posterior, llavors l'hem d'esborrar, perquè gairebé segur que està incomplet (en canvi, podem donar per bo l'últim arxiu XSN-..., que s'haurà pogut completar).

Abordarem aquesta recapitulació des de l'òptica del desencís, és a dir, sabent que la pretensió de constituir un inventari complet de SUDOKUs-SOLUCIÓ normalitzades és irrealitzable, però amb la satisfacció relativa de haver millorat unes eines que, malgrat que segueixen sent inservibles de cara a la consecució d'un objectiu tan utòpic, estan més esmolades i poden tenir alguna utilitat en feines auxiliars: al cap i a la fi, encara que amb l'ajut de la supercomputació haguéssim assolit un inventari tan preuat com difícil d'emmagatzemar i consultar, l'autor ha descobert (i el lector tindrà ocasió de fer-ho també, amb l'únic requisit de tenir paciència per empassar-se aquest document fins a l'última gota) que la pressumpció inicial que la part problemàtica consistia a disposar d'una SUDOKU-SOLUCIÓ, perquè obtenir SUDOKUs-PROBLEMA a partir d'aquesta estava pastat, era una falòrnia com una casa.

Us servirem el codi aprofitable d'aquesta aventura distribuït en tres paquets, el primer dels quals fa referència a l'arxiu de línies 123456789.txt: per crear-lo, per llegir-lo, convertint-lo en la llista **L1**, i per saber quina posició ocupa en aquesta llista una determinada fila; hem prescindit de la funció **LOC1A9L1**, perquè aquells valors de la llista **LOC** que ens podien interessar ja els hem incorporat directament en les funcions que els requerien i perquè **FES-KOMPAT-3-1** (a la segona versió) també n'havia prescindit. Sobre **C:CARREGA-123456789**, destacarem el fet que l'haguem mantingut com a ordre d'AutoCAD (malgrat que s'hi accedeixi exclusivament des d'altres funcions) en atenció a la comoditat del'usuari, que podria necessitar executar-la per disposar de la llista **L1** com a variable global en tota la sessió de treball, i així tener accés a **LOCL1**; si haguéssim integrat **(CARREGA-123456789)** en **LOCL1** hauríem pogut deixar-la com a funció ordinària, però l'espera d'un segon en una funció que caldrà executar sovint (quan l'usuari, per exemple, vulgui saber quines posicions tenen certes files a **L1**, per poder-les utilitzar com a argument de **FES-KOMPAT-3-1**), podria acabar resultant molesta. Aquí teniu el primer paquet:

```
(defun C:FES-123456789 (/ A M N)
  (setq A (open "123456789.txt" "w")
        M 123456788)
  (repeat 864197533
    (if (wcmatch (setq M (1+ M) N (itoa M)) "*0*")
        (setq N ())
        (foreach C '("1" "2" "3" "4" "5" "6" "7" "8" "9")
          (if (and N (not (wcmatch N (strcat "*" C "*")))) (setq N ()))))
    (if N (write-line N A)))
  (close A))

(defun C:CARREGA-123456789 (/ A)
  (setq RUTA (findfile "123456789.txt") A (open RUTA "r") L1 ())
  (repeat 362880 (setq L1 (cons (read-line A) L1)))
  (setq L1 (reverse L1))
  (close A))

(defun LOCL1 (E) (- 362880 (length (member E L1))))
```

Tot seguit veureu les funcions relacionades amb la construcció d'arxius KOMPAT-... de tríades normalitzades, tasca que podreu abordar de dues maneres: en la línia de la funció **FES-COMPAT-3-1** (limitada a la construcció d'un arxiu COMPAT-... i única efectivament operativa en l'etapa fundacional d'aquest treball) i també en la de **FES-COMPAT-M-N-O** (amb capacitat per fer una sèrie ordenada i gairebé il·limitada d'arxius, capacitat que esdevé més versemblant en canviar al format KOMPAT-...).

Abans d'oferir-vos aquest bloc de codi, però, convé aclarir per què hem modificat la funció bàsica **FES-KOMPAT-3-1** (que, en la línia de dotar totes les funcions de certa coherència lèxica, passem a anomenar **FES-TRIADES**). Com que ara està pensada per ser accedida des de **C:FES-TRIADES** i **C:FES-STRIADES**, on farem les assignacions
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)))
i **(setq A (open (strcat RUTA "KOMPAT-" (itoa M) ".txt") "w"))**, que treiem d'aquí, introduïm la variable **F** per evitar confusions i la funció queda més o menys així:

```
(defun FES-TRIADES (/ F I I1 I2 I3 J K L N FM FN LTM RLTM)
  (setq FM (nth M L1)
        I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
        LTM (list I3 I2 I1) J 3)
  (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
  (setq RLTM (reverse LTM))
  (if (> I1 I2) (setq I I1 I1 I2 I2 I))
  (if (> I2 I3)
      (progn
        (setq I I2 I2 I3 I3 I)
        (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
  (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
        K (* (/ M 40320) 40320))
  (while (< (setq K (+ K 40320)) 362880)
         (foreach J I (if (= K (* J 40320)) (setq K (+ K 40320))))
         (if (< K 362880)
             (progn
               (setq N (1- K) I1 (+ K 40320))
               (while (< (setq N (1+ N)) I1)
                      (setq FN (nth N L1))
                      (if (COMPAT) (setq L (cons N L)))))))
  (if L
      (progn
        (setq F (open A "w"))
        (foreach E (reverse L) (write-line (itoa E) F))
        (close F))))
```

Ara bé, el molt lloable propòsit d'impedir la creació d'arxius KOMPAT-... buits, construint abans una llista **L** amb les segones files compatibles amb la primera de la tríada, comú a totes elles, i condicionar la creació de l'arxiu a l'existència efectiva de components, ha xocat amb la també lloable intenció de no complicar més el disseny de la versió ampliada de **FES-SUDOKUS** que hem anunciat: la inviabilitat de tríades normalitzades encapçalades per determinades files, i el fet que alguns d'aquests casos es traduïssin en la inexistència d'arxiu (precisament per l'acció del dispositiu esmentat) i d'altres ho fessin en arxius poblats de força registres (per bé que, rascant-ne el contingut, acabéssim descobrint que no servien de res), ens hauria dut força maldecaps i, atès que la via encetada en aquest capítol no té massa futur, semblava més realista reduir dificultats fent que totes les files amb dret a disposar d'arxiu KOMPAT-... per la seva posició a **L1** (les primeres 282.240) en tinguessin un, tot i que aquest arxiu no fos més que un nom buit de contingut; per això, la versió de **FES-TRIADES** que hem adoptat per ser compatible amb la resta de funcions és la que teniu més endavant, després de la funció subordinada **COMPAT**. De tota manera, igual que **COMPAT** no és accedida directament per l'usuari sinó que ho és des de **FES-TRIADES**, aquesta tampoc no ho serà, sinó que serà compartida per dues funcions alternatives, ambdues subministrades en forma d'ordres d'AutoCAD per a una major comoditat de l'usuari: **C:FES-TRIADES** + **FES-TRIADES** és una adaptació de **FES-KOMPAT-3-1** que, en comptes de requerir el previ alliçonament de l'usuari en el subministrament de l'argument, l'hi demana; **C:FES-STRIADES** + **FES-TRIADES** seria una translació de **FES-COMPAT-M-N-O** al format KOMPAT-... que, com l'original, permet de crear una sèrie ordenada d'aquests arxius, tan llarga com l'usuari vulgui o pugui, demanant-li la posició en **L1** de la fila corresponent a l'arxiu de tríades inicial. No desaprofitarem l'ocasió de llençar una galleda d'aigua freda sobre tots aquells que hagin dipositat unes expectatives massa optimistes sobre algunes expressions que no pretenien suscitar-les (almenys en termes absoluts), com "capacitat per crear una sèrie ordenada i gairebé il·limitada d'arxius, capacitat que esdevé més versemblant en canviar al format KOMPAT-...": suposant que l'ocupació mitjana dels 282.240 arxius KOMPAT-... que necessitaríem per inventariar totes les solucions normalitzades fos de **50 KB** (KOMPAT-0.txt ocupa 95 KB i KOMPAT-281519.txt, que és l'arxiu no nul més elevat, n'ocupa 16) i trigués **7'30"** a ser construït, en total ocuparien uns **14 GB** i trigarien uns **4 anys** a completar-se. Val a dir que aquests temps desorbitats responen a l'equip carregat d'anys usat per l'autor, però fins i

tot acceptant que un PC actual (referint-nos a la data de publicació del present treball) pogués reduir-los a la meitat, els resultats no són per saltar d'alegria. Ja sense més preàmbuls, les funcions implicades en la creació d'arxius KOMPAT-...:

```
(defun 3*3 (J / LM LN)
  (repeat 3 (setq J (1+ J) LM (cons (nth J LTM) LM) LN (cons (nth J LTN) LN)))
  (not (or (member (car LN) LM) (member (cadr LN) LM) (member (last LN) LM))))

(defun COMPAT (/ LTN COMP TMJ TNJ)
  (setq LTN (list (substr FN 1 1)) COMP T J 1)
  (while (and COMP (< J 9))
         (setq TMJ (nth J RLTM) J (1+ J) TNJ (substr FN J 1)
               COMP (/= TMJ TNJ))
         (if COMP (setq LTN (cons TNJ LTN))))
  (and COMP (3*3 -1) (3*3 2) (3*3 5)))

(defun FES-TRIADES (/ F I I1 I2 I3 J K L N FM FN LTM RLTM)
  (setq F (open A "w") FM (nth M L1)
        I1 (substr FM 1 1) I2 (substr FM 2 1) I3 (substr FM 3 1)
        LTM (list I3 I2 I1) J 3)
  (while (< J 9) (setq J (1+ J) LTM (cons (substr FM J 1) LTM)))
  (setq RLTM (reverse LTM))
  (if (> I1 I2) (setq I I1 I1 I2 I2 I))
  (if (> I2 I3)
      (progn
        (setq I I2 I2 I3 I3 I)
        (if (> I1 I2) (setq I I1 I1 I2 I2 I))))
  (setq I (list (1- (atoi I1)) (1- (atoi I2)) (1- (atoi I3)))
        K (* (/ M 40320) 40320))
  (while (< (setq K (+ K 40320)) 362880)
         (foreach J I (if (= K (* J 40320)) (setq K (+ K 40320))))
         (if (< K 362880)
             (progn
               (setq N (1- K) I1 (+ K 40320))
               (while (< (setq N (1+ N)) I1)
                      (setq FN (nth N L1))
                      (if (COMPAT) (write-line (itoa N) F))))))
  (close F))

(defun C:FES-TRIADES (/ RUTA L1 M A)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)))
  (while (progn
           (setq M (fix (getreal
                          (strcat "\n.\nPer crear un arxiu amb les tríades norma"
                                  "litzades que tenen una mateixa 1ª fila"
                                  "\nheu de donar la seva posició en la llista "
                                  "ordenada de totes les files possibles"
                                  "\n(valors entre 0 i 282239): "))))
           (or (< M 0) (> M  282239)
               (findfile (setq A (strcat RUTA "KOMPAT-" (itoa M) ".txt")))))
         (if (or (< M 0) (> M  282239))
             (prompt (strcat "\n.\nEl valor introduït se situa fora dels límits "
                             "assenyalats."))
             (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt ja hi "
                             "ha un arxiu KOMPAT-" (itoa M) ".txt.")))
         (prompt "\n                      [Per repetir, polseu qualsevol tecla]")
         (grread))
  (FES-TRIADES)
  (princ))

(defun C:FES-STRIADES (/ RUTA L1 M A)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)))
```

```
(while (progn
         (setq M (fix (getreal
                        (strcat "\n.\nPer crear una sèrie d'arxius de tríades "
                                "normalitzades (cada arxiu amb les que"
                                "\ntenen una mateixa 1ª fila) cal donar la "
                                "posició de la 1ª fila del primer arxiu"
                                "\nen la llista ordenada de files possibles "
                                "(valors entre 0 i 282239): "))))
         (or (< M 0) (> M 282239)))
       (prompt "\n.\nEl valor introduït se situa fora dels límits assenyalats.")
       (prompt "\n                    [Per repetir, polseu qualsevol tecla]")
       (grread))
(setq M (1- M))
(while (< (setq M (1+ M)) 282240)
       (if (not (findfile (setq A (strcat RUTA "KOMPAT-" (itoa M) ".txt"))))
           (FES-TRIADES)))
(princ))
```

Per acabar, proseguirem amb les funcions que ens permetran crear arxius XSN-... de solucions normalitzades (un a un o per sèries ordenades) i veure-les en pantalla. Si a la màquina de fabricar l'arxiu XSN-... que emmagatzema el conjunt de SUDOKUs-SOLUCIÓ corresponent a les tríades que comencen per unes files 1, 4 i 7 prefixades l'anomenàvem **FES-SUDOKUS**, a la que teòricament pot fabricar una sèrie il·limitada d'arxius (posats a somiar, podríem començar per XSN-0-45507-80884.txt i arribar a XSN-40319-156095-270716.txt, si l'execució no fos interrompuda voluntàriament per l'usuari o l'equip col·lapsés per manca d'espai en disc o fallada de components, situació idíl·lica en què hauríem assolit totes les solucions normalitzades i on només faltaria permutar les files dels requadres 9×3 o permutar aquests requadres per obtenir totes les SUDOKUs-SOLUCIÓ) hem pensat a posar-li de nom **FES-SSUDOKUS**. D'acord que no ens hem trencat el cap buscant un nom més adient i ens hem limitat a afegir la **S** de **S**èrie, igual que per passar de **FES-TRIADES** a **FES-STRIADES**, però és que tampoc volíem contrarrestar a nivell formal allò que havia quedat a mitges a nivell de contingut: no preteníem altra cosa que crear una funció que funcionés. I encara rai que n'hem modificat l'estructura per evitar repeticions en el conjunt del codi, com apreciareu tot seguit, perquè tot es feia des de la consciència que aquesta era una batalla perduda (en el sentit que la pretensió d'inventariar les solucions duia a un cul-de-sac, com diu el títol) i que era absurd esmerçar més temps en augmentar l'eficiència d'un engranatge que, com a sistema global, estava condemnat al fracàs, encara que ocasionalmentse se li pogués treure algun profit. Insistim en això perquè, si la línia de treball hagués tingut més possibilitats, no hagués estat admissible deixar amb tantes incoherències la gestió dels arxius KOMPAT-...: recordareu que en la substitució del format COMPAT-... pel KOMPAT-... no tot eren flors i violes, sinó que l'indiscutible pas endavant (en ocupació i en temps) el donàvem a costa de la feina assignada a **CARREGA-TRIADES** < **FES-SUDOKUS**, que no només passava per restituir les terceres files compatibles de cada tríada, sinó que incloïa l'obligació prèvia de desestimar les segones files que tampoc no n'hauríen trobat de compatibles que alhora complissin els requisits propis d'una solució normalitzada, podent-se donar el cas que en aquesta depuració no se salvés cap registre de la crema (no parlem, doncs, d'aquells arxius KOMPAT-... que amb l'última versió de **FES-TRIADES** hem fet nèixer buits per no complicar **FES-SSUDOKUS**, adaptant-la a l'eventualitat que determinats arxius de tríades no existissin ni poguessin existir com a tals, sinó dels que haurien estat buits en l'antic format COMPAT-... però no hi queden en el KOMPAT-..., tot i ser de més ràpida creació). Pel que fa a l'ús ocasional de **FES-SSUDOKUS** en la consecució, per exemple, d'uns quants arxius XSN-... correlatius (o gairebé), per obtenir un grapat de solucions pròximes entre les quals poder triar la més adient per treure un SUDOKU-PROBLEMA, sí que us podem garantir que la intervenció d'aquests arxius KOMPAT-... fantasmes (dels de contingut irrellevant i dels buits de naixement) no provocarà cap error d'execució ni la interrupció del procés: només es generaran els també fantasmes XSN-... buits, com podeu comprovar substituint la 7ª fila del primer conjunt de solucions normalitzades (XSN-0-45507-80884, presentat en un incís al començament d'aquest mateix paràgraf) per la fila "697812345" (de posició 241.080 a **L1** i que correspon a l'arxiu KOMPAT-241080.txt, força petit però no buit), o la del conjunt XSN-39600-46233-92464 (amb què havíem estat jugant abans) per la fila "789653412" (de posició 277.192 a **L1** i que correspon a l'arxiu buit KOMPAT-277192.txt), veient que ambdós arxius resultants, XSN-0-45507-241080.txt i XSN-39600-46233-277192.txt, són de longitud nul·la. Aclarit això, comentarem per sobre els canvis introduïts.

Pel que fa a **C:FES-SUDOKUS** i **C:FES-SSUDOKUS** (com veieu, també les servim en forma d'ordres d'AutoCAD), n'hem segregat la segona meitat (els sis bucles niats, amb la inicialització i conclusió) per poder-la compartir, amb el nom de funció **6BUCLES**, i en **CARREGA-TRIADES** hem hagut d'incorporar **TF1** i **TF2** com a variables locals, atès que les homònimes de **C:FES-SSUDOKUS** representaran el paper de senyals binaris que, en passar del conjunt de tríades inicial (definit per l'usuari) a la dinàmica que hem avançat quatre fulls enrera, faran que les successives combinacions de tríades a considerar es caracteritzin únicament per la variació de la tercera (encapçalada per la 7ª fila, de posició **M3**), mantenint la segona (encapçalada per la 4ª fila, de posició **M2**) mentre **M3** no sobrepassi la posició 282.239, i mantenint la primera (encapçalada per la 1ª fila, de posició **M1**) mentre **M2** no sobrepassi la posició 161.279. De la resta de funcions subordinades (**STRCAT-1** a **C:FES-SUDOKUS**, **STRCAT-2** a **C:FES-SSUDOKUS** i **FES-LIMI** a ambdues) no cal aclarir res d'especial, tret potser de **TRIADES**, que es refereix a una de les singularitats de **C:FES-SSUDOKUS** respecte a **C:FES-SUDOKUS**: la capacitat de crear en temps real els arxius KOMPAT-... que es precisen però no s'han construït prèviament, que en el cas de la 1ª i 4ª fila de l'arxiu inicial aconsellen advertir l'usuari sobre la inevitable espera abans de seguir amb la introducció de les dades pendents (posicions de la 4ª i/o 7ª fila).

Com abans, i recordant que **C:FES-SUDOKUS** es limita a construir l'arxiu XSN-... amb les solucions normalitzades que comparteixen la 1ª, 4ª i 7ª fila corresponents als tres arxius KOMPAT-... especificats (que prèviament hem hagut de crear, sigui amb **C:FES-TRIADES** o amb **C:FES-STRIADES**), que **C:FES-SSUDOKUS** permet de construir-ne una sèrie, a partir d'un conjunt inicial de solucions normalitzades (tant se val que els arxius KOMPAT-... corresponents a aquest primer conjunt especificat i als que vindran després existeixin o no), i que **C:VIS-XSN** ens en permet la visualització, presentarem totes les funcions implicades, tant si són noves, han estat afectades per alguna reestructuració o canvi de nom, o romanen com a la primera presentació:

```
(defun STRCAT-1 (T1 T2 T3)
  (strcat "\n.\nEntreu el nom d'un arxiu de tríades compatibles existent que "
          "correspongui a la\n" T1 "a tríada de la SUDOKU-SOLUCIÓ (de KOMPAT-"
          T2 ".txt a KOMPAT-" T3 ".txt),\nsense l'extensió .txt: KOMPAT-"))

(defun RESTA (T1 T2 / T21 LC)
  (setq T21 (strcat T2 T1))
  (foreach C C9A1
    (if (not (wcmatch T21 (strcat "*" C "*"))) (setq LC (cons C LC))))
  LC)

(defun CAT (C1 C2 C3) (strcat (nth C1 LC) (nth C2 LC) (nth C3 LC)))

(defun PERM (LC COMP / LT3 LT OK)
  (setq LT3 (list (CAT 0 1 2))
        LT3 (cons (CAT 0 2 1) LT3)
        LT3 (cons (CAT 1 0 2) LT3)
        LT3 (cons (CAT 1 2 0) LT3)
        LT3 (cons (CAT 2 0 1) LT3)
        LT3 (cons (CAT 2 1 0) LT3))
  (if COMP
      (progn
        (setq OK T)
        (foreach T3 LT3
          (if OK
              (if (< (substr T3 1 1) COMP)
                  (setq OK ())
                  (setq LT (cons T3 LT)))))
         LT)
      (reverse LT3)))

(defun CARREGA-TRIADES (A M / TF1-1 TF1-2 TF1-3 TF2-1 TF2-2 TF2-3 TF3-1 TF3-2
                              TF3-3 LC3-1 LC3-2 LC3-3 LTF3 LTF23 LTF123 TF1 TF2)
  (setq A (open A "r")
        TF1 (nth M L1)
        TF1-1 (substr TF1 1 3) TF1-2 (substr TF1 4 3) TF1-3 (substr TF1 7 3))
  (while (setq TF2 (read-line A))
         (setq TF2 (nth (atoi TF2) L1) TF2-1 (substr TF2 1 3)
               LC3-1 (RESTA TF1-1 TF2-1) LTF3-1 (PERM LC3-1 (substr TF2-1 1 1)))
```

```
          (if LTF3-1
              (progn
                (setq TF2-2 (substr TF2 4 3) TF2-3 (substr TF2 7 3)
                      LC3-2 (RESTA TF1-2 TF2-2) LTF3-2 (PERM LC3-2 ())
                      LC3-3 (RESTA TF1-3 TF2-3) LTF3-3 (PERM LC3-3 ())
                      LTF23 () LTF3 ())
                (foreach TF3-1 LTF3-1
                  (foreach TF3-2 LTF3-2
                    (foreach TF3-3 LTF3-3
                      (setq LTF3 (cons (strcat TF3-1 TF3-2 TF3-3) LTF3)))))
                (setq LTF23 (list TF2 (reverse LTF3))
                      LTF123 (cons LTF23 LTF123)))))
  (close A)
  (cons TF1 (reverse LTF123)))

(defun FES-LIMI ()
  (if (= (substr TF2 1 1) "2")
      "80640"
      (if (= (substr TF2 1 1) "3") "120960" "161280")))

(defun BLANCS (A B / CC)
  (setq CC "")
  (repeat (- (strlen (itoa A)) (strlen (itoa B)))
    (setq CC (strcat CC " ")))
  (if CC CC ""))

(defun COMP (LF / PERF P)
  (setq K 0 PERF "")
  (repeat 9
    (setq K (1+ K) P "")
    (foreach F LF (setq P (strcat P (substr F K 1))))
    (setq PERF (strcat PERF "[~" P "]"))))

(defun REST-1 (C REST / CC)
  (foreach R (reverse REST)
    (if (/= R C) (setq CC (cons R CC))))
  CC)

(defun 6BUCLES (/ PERF1 T1F2 LT1F3 PERF2 T2F2 LT2F3 PERF3 T3F2 T3F3 REST9)

  (setq LT3 (CARREGA-TRIADES A3 M3) T3F1 (car LT3) LT3F23 (cdr LT3)
        PERF1 (COMP (list T2F1 T3F1))
        A3 (open (strcat RUTA "XSN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt")
                 "w"))
  (foreach T1F23 LT1F23
    (setq T1F2 (car T1F23))
    (if (wcmatch T1F2 PERF1)
        (progn
          (setq LT1F3 (cadr T1F23))
          (foreach T1F3 LT1F3
            (if (wcmatch T1F3 PERF1)
                (progn
                  (setq PERF2 (COMP (list T1F1 T1F2 T1F3 T3F1)))
                  (foreach T2F23 LT2F23
                    (setq T2F2 (car T2F23))
                    (if (wcmatch T2F2 PERF2)
                        (progn
                          (setq LT2F3 (cadr T2F23))
                          (foreach T2F3 LT2F3
                            (if (wcmatch T2F3 PERF2)
                                (progn
                                  (setq PERF3 (COMP (list T1F1 T1F2 T1F3 T2F1 T2F2
                                                          T2F3)))
                                  (foreach T3F23 LT3F23
                                    (setq T3F2 (car T3F23))
                                    (if (wcmatch T3F2 PERF3)
                                        (progn
                                          (setq LT3F3 (cadr T3F23) OK T)
```

```
                                          (foreach T3F3 LT3F3
                                            (if (and OK (wcmatch T3F3 PERF3))
                                                (progn
                                                  (setq OK ())
                                                  (write-line
                                                    (strcat (substr T1F2 1 8)
                                                            (substr T1F3 1 8)
                                                            (substr T2F2 1 8)
                                                            (substr T2F3 1 8)
                                                            (substr T3F2 1 8))
                                                    A3))))))))))))))))))))
  (close A3)
  (princ))

(defun C:FES-SUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1 OK2
                        LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1 LT3F23)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)) A1 T A2 T A3 T
        C9A1 '("9" "8" "7" "6" "5" "4" "3" "2" "1"))
  (while (progn
           (setq M1 (fix (getreal (STRCAT-1 "primer" "0" "40319"))))
           (not (and (>= M1 0) (<= M1 40319)
                     (setq A1 (findfile (strcat RUTA "KOMPAT-" (itoa M1)
                                                ".txt"))))))
         (if (not A1)
             (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no hi "
                             "ha cap arxiu KOMPAT-" (itoa M1) ".txt."))
             (prompt (strcat "\n.\nEl valor introduït se situa fora dels límits "
                             "assenyalats.")))
         (prompt "\n                      [Per repetir, polseu qualsevol tecla]")
         (grread))
  (setq TF1 (nth M1 L1))
  (while (progn
           (setq M2 (fix (getreal (STRCAT-1 "segon" "40320" "161279")))
                 TF2 (nth M2 L1))
           (not (and (>= M2 40320) (<= M2 161279)
                     (setq K 0 OK T A2 (findfile (strcat RUTA "KOMPAT-" (itoa M2)
                                                         ".txt")))
                     (progn
                       (repeat 9
                         (if OK (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                             (substr TF2 K 1)))))
                       OK))))
         (if (not A2)
             (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no hi "
                             "ha cap arxiu KOMPAT-" (itoa M2) ".txt."))
             (if OK
                 (prompt (strcat "\n.\nEl valor introduït se situa fora dels "
                                 "límits assenyalats."))
                 (prompt (strcat "\n.\nLa fila " (itoa M2) "ª (" TF2 ") és "
                                 "incompatible\n" (BLANCS M2 M1) " amb la "
                                 (itoa M1) "ª (" TF1 ")."))))
         (prompt "\n                        [Per repetir, polseu qualsevol tecla]")
         (grread))
  (setq LIMI (FES-LIMI))
  (while (progn
           (setq M3 (fix (getreal (STRCAT-1 "tercer" LIMI "282239"))))
           (not (and (>= M3 (atoi LIMI)) (<= M3 282239)
                     (setq A3 (findfile (strcat RUTA "KOMPAT-" (itoa M3) ".txt")))
                     (progn
                       (setq TF3 (nth M3 L1) K 0 OK1 T OK2 T)
                       (repeat 9 (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                              (substr TF3 K 1))
                                       OK1 (if OK OK1 ())
                                       OK (/= (substr TF2 K 1) (substr TF3 K 1))
                                       OK2 (if OK OK2 ()))))
```

```
                        (setq OK (and OK1 OK2))))))
        (if (not A3)
            (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no hi "
                            "ha cap arxiu KOMPAT-" (itoa M3) ".txt."))
            (if OK
                (prompt (strcat "\n.\nEl valor introduït se situa fora dels "
                                "límits assenyalats."))
                (prompt (strcat "\n.\nLa fila " (itoa M3) "ª (" TF3 ") és "
                                "incompatible\n"
                                (if (not OK2)
                                    (strcat (BLANCS M3 M2) " amb la " (itoa M2)
                                            "ª (" TF2 ")" (if OK1 "." " i\n"))
                                    "")
                                (if (not OK1)
                                    (strcat (BLANCS M3 M1) " amb la " (itoa M1)
                                            "ª (" TF1 ").")
                                    ""))))))
        (prompt "\n                    [Per repetir, polseu qualsevol tecla]")
        (grread))
  (setq LT1 (strcat "XSN-" (itoa M1) "-" (itoa M2) "-" (itoa M3) ".txt"))
  (if (findfile (strcat RUTA LT1))
      (prompt (strcat "\n.\nL'arxiu " LT1 " ja existeix!"))
      (progn
        (setq LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
              LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2))
        (6BUCLES)))
  (princ))

 (defun STRCAT-2 (T1 T2 T3 T4)
   (strcat (if (= (substr (getvar "LASTPROMPT") 1 5) "sense") "" "\n")
           ".\nNom de l'arxiu de tríades compatibles corresponent a la " T1
           "ª tríada del primer\narxiu eX" T2 " de Solucions Normalitzades "
           "(de KOMPAT-" T3 ".txt a KOMPAT-" T4 ".txt),\nsense l'extensió .txt: "
           "KOMPAT-"))

(defun TRIADES (M AVIS / A)
  (if (not (findfile (setq A (strcat RUTA "KOMPAT-" (itoa M) ".txt"))))
      (progn
        (if AVIS
            (progn
              (prompt "(espereu uns minuts, mentre es construeix)")
              (terpri)))
        (FES-TRIADES)))
  A)

(defun C:FES-SSUDOKUS (/ L1 RUTA C9A1 A1 A2 A3 M1 M2 M3 TF1 TF2 TF3 LIMI OK OK1
                         OK2 LT1 T1F1 LT1F23 LT2 T2F1 LT2F23 LT3 T3F1 LT3F23
                         T1 T2)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)) A1 T A2 T A3 T
        C9A1 '("9" "8" "7" "6" "5" "4" "3" "2" "1"))
  (prompt "\n.\nPer crear una sèrie d'arxius de Solucions Normalitzades ")
  (prompt "cal donar el nom dels\n3 arxius de tríades que necessita el primer ")
  (prompt "(si no existeixen, seran creats).")
  (prompt "\n                      [Per seguir, polseu qualsevol tecla]")
  (grread)
  (while (progn
           (setq M1 (fix (getreal (STRCAT-2 "1" "plícit" "0" "40319"))))
           (not (and (>= M1 0) (<= M1 40319))))
         (prompt "\n.\nEl valor introduït se situa fora dels límits assenyalats.")
         (prompt "\n                      [Per repetir, polseu qualsevol tecla]")
         (grread))
  (setq A1 (TRIADES M1 T) TF1 (nth M1 L1))
  (while (progn
           (setq M2 (fix (getreal (STRCAT-2 "2" "." "40320" "161279")))
                 TF2 (nth M2 L1) K 0 OK T)
```

```
          (not (and (>= M2 40320) (<= M2 161279)
                    (progn
                      (repeat 9
                        (if OK (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                            (substr TF2 K 1)))))
                       OK))))
        (if OK
            (prompt (strcat "\n.\nEl valor introduït se situa fora dels límits "
                            "assenyalats."))
            (prompt (strcat "\n.\nLa fila " (itoa M2) "ª (" TF2 ") és "
                            "incompatible\n" (BLANCS M2 M1) " amb la " (itoa M1)
                            "ª (" TF1 ").")))
        (prompt "\n              [Per repetir, polseu qualsevol tecla]")
        (grread))
(setq A2 (TRIADES M2 T) LIMI (FES-LIMI))
(while (progn
          (setq M3 (fix (getreal (STRCAT-2 "3" "." LIMI "282239")))
                TF3 (nth M3 L1) K 0 OK1 T OK2 T)
          (not (and (>= M3 (atoi LIMI)) (<= M3 282239)
                    (progn
                      (repeat 9 (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                             (substr TF3 K 1))
                                      OK1 (if OK OK1 ())
                                      OK (/= (substr TF2 K 1) (substr TF3 K 1))
                                      OK2 (if OK OK2 ())))
                      (setq OK (and OK1 OK2))))))
        (if OK
            (prompt (strcat "\n.\nEl valor introduït se situa fora dels límits "
                            "assenyalats."))
            (prompt (strcat "\n.\nLa fila " (itoa M3) "ª (" TF3 ") és "
                            "incompatible\n"
                            (if (not OK2)
                                (strcat (BLANCS M3 M2) " amb la " (itoa M2)
                                        "ª (" TF2 ")" (if OK1 "." " i\n"))
                                "")
                            (if (not OK1)
                                (strcat (BLANCS M3 M1) " amb la " (itoa M1)
                                        "ª (" TF1 ").")
                                ""))))
        (prompt "\n                      [Per repetir, polseu qualsevol tecla]")
        (grread))
(setq A3 (TRIADES M3 ())
      LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)
      LT2 (CARREGA-TRIADES A2 M2) T2F1 (car LT2) LT2F23 (cdr LT2))
(if (not (findfile (strcat RUTA "XSN-" (itoa M1) "-" (itoa M2) "-" (itoa M3)
                           ".txt")))
    (6BUCLES))
(while (or (not T1) (< (setq M1 (1+ M1)) 40320))
        (if T1
            (setq A1 (TRIADES M1 ()) TF1 (nth M1 L1)
                  LT1 (CARREGA-TRIADES A1 M1) T1F1 (car LT1) LT1F23 (cdr LT1)))
        (while (or (not T2) (< (setq M2 (1+ M2)) 161280))
               (if T2
                   (progn
                     (setq TF2 (nth M2 L1) K 0 OK T)
                     (repeat 9 (if OK (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                                   (substr TF2 K 1)))))
                     (if OK
                         (setq A2 (TRIADES M2 ())
                               LT2 (CARREGA-TRIADES A2 M2)
                               T2F1 (car LT2) LT2F23 (cdr LT2)))))
               (while (< (setq M3 (1+ M3)) 282240)
                      (setq TF3 (nth M3 L1) K 0 OK1 T OK2 T)
                      (repeat 9 (setq OK (/= (substr TF1 (setq K (1+ K)) 1)
                                             (substr TF3 K 1))
                                      OK1 (if OK OK1 ())
                                      OK (/= (substr TF2 K 1) (substr TF3 K 1))
                                      OK2 (if OK OK2 ())))
```

```
                        (if (and OK1 OK2)
                            (progn
                              (setq A3 (TRIADES M3 ()))
                              (if (not (findfile (strcat RUTA "XSN-" (itoa M1) "-"
                                                         (itoa M2) "-" (itoa M3)
                                                         ".txt")))
                                        (6BUCLES)))))
                  (setq LIMI (FES-LIMI) M3 (1- (atoi LIMI)) T2 T))
          (setq M2 40319 T1 T))
  (princ))

(defun COMPLETA (F)
  (setq J 0 OK T L ())
  (repeat 8 (setq J (1+ J) L (cons (substr F J 1) L)))
  (foreach C 9C
    (if (and OK (not (member C L)))
        (setq OK () L (strcat F C))))
  L)

(defun PREFORM (/ OK L 9C)
  (setq 9C '("1" "2" "3" "4" "5" "6" "7" "8" "9")
        F2 (substr F 1 8) F2 (COMPLETA F2)
        F3 (substr F 9 8) F3 (COMPLETA F3)
        F5 (substr F 17 8) F5 (COMPLETA F5)
        F6 (substr F 25 8) F6 (COMPLETA F6)
        F8 (substr F 33 8) F8 (COMPLETA F8)
        F9 "" J 0)
  (repeat 9
    (setq J (1+ J) OK T L ())
    (foreach F (list F1* F2 F3 F4* F5 F6 F7* F8)
      (setq L (cons (substr F J 1) L)))
    (foreach C 9C
      (if (and OK (not (member C L))) (setq OK () F9 (strcat F9 C))))))

(defun FORMAT (L / F FT)
  (foreach QF L
    (setq F (eval QF) FT "   " J 0)
    (repeat 3
      (setq FT (strcat FT " "))
      (repeat 3 (setq J (1+ J) FT (strcat FT " " (substr F J 1)))))
    (set QF FT)))

(defun IMPR (F1 F2 F3 F4 F5 F6 F7 F8 F9)
  (terpri)
  (terpri) (princ F9) (terpri) (princ F8) (terpri) (princ F7) (terpri)
  (terpri) (princ F6) (terpri) (princ F5) (terpri) (princ F4) (terpri)
  (terpri) (princ F3) (terpri) (princ F2) (terpri) (princ F1) (terpri))

(defun C:VIS-XSN (/ L1 RUTA A A1 A1P F I J K O F1 F2 F3 F4 F5 F6 F7 F8 F9
                    F1* F4* F7* F31 F32 F33 F34 F35 F36 F37 F38 F39)
  (textscr)
  (C:CARREGA-123456789)
  (repeat 40 (terpri))
  (setq RUTA (substr RUTA 1 (- (strlen RUTA) 13)))
  (while (progn
           (setq A (getstring (strcat "\n.\nEntreu el nom d'algun arxiu eXplícit "
                                      "de Solucions Normalitzades\n(XSN-pF1-pF4-"
                                      "pF7), sense l'extensió .txt: XSN-")))
           (not (setq A1 (findfile (strcat RUTA "XSN-" A ".txt")))))
         (prompt (strcat "\n.\nEn el directori on teniu 123456789.txt no s'hi ha "
                         "trobat cap arxiu \nXSN-" A ".txt. REPETIU!")))
  (setq A1P (open A1 "r")
        F A F1 (itoa (atoi F))
        F (substr F (+ (strlen F1) 2)) F4 (itoa (atoi F))
        F (substr F (+ (strlen F4) 2)) F7 (atoi F)
        F1 (nth (atoi F1) L1) F4 (nth (atoi F4) L1) F7 (nth F7 L1)
        F1* F1 F4* F4 F7* F7 K 0)
  (FORMAT (list 'F1 'F4 'F7))
```

```
  (while (read-line A1P) (setq K (1+ K)))
  (close A1P)
  (terpri)
  (prompt (strcat "\nL'arxiu XSN-" A ".txt té " (itoa K)
                  " solucions normalitzades:"))
  (prompt "\n1) Podeu veure-les totes seqüencialment, de 3 en 3.")
  (prompt (strcat "\n2) Podeu veure-les reclamant-les per la seva posició "
                  "(1 a " (itoa K) ").\n"))
  (while (progn
           (setq O (getint "Opció 1 o 2: "))
           (or (< O 1) (> O 2)))
         (prompt "\n  REPETIU! "))
  (if (= O 1)
      (progn
        (setq A1P (open A1 "r") I 3)
        (while (and (= I 3) (> (setq I (if (> (+ O 2) K) (rem K 3) 3)) 0))
               (prompt "\n                    Per seguir, polseu qualsevol tecla ")
               (prompt "(<Esc> per acabar)")
               (grread)
               (prompt (if (> I 1)
                           (strcat "\nSolucions " (itoa O) " "
                             (if (> I 2) "a" "i") " " (itoa (+ O (1- I))) ":")
                           (strcat "\n Solució " (itoa O) ":")))
               (setq F31 "" F32 "" F33 "" F34 "" F35 "" F36 "" F37 "" F38 ""
                     F39 "")
               (repeat I
                 (setq O (1+ O) F (read-line A1P))
                 (PREFORM)
                 (FORMAT (list 'F2 'F3 'F5 'F6 'F8 'F9))
                 (setq F31 (strcat F31 F1) F32 (strcat F32 F2) F33 (strcat F33 F3)
                       F34 (strcat F34 F4) F35 (strcat F35 F5) F36 (strcat F36 F6)
                       F37 (strcat F37 F7) F38 (strcat F38 F8)
                       F39 (strcat F39 F9)))
               (IMPR F31 F32 F33 F34 F35 F36 F37 F38 F39))
        (close A1P))
      (while F
             (setq A1P (open A1 "r"))
             (terpri)
             (while (progn
                      (setq F (getreal (strcat "Entreu núm. solució, entre 1 i "
                                               (itoa K) ", "
                                               "(<CR> o <blanc> per acabar): ")))
                      (and F (or (< F 1) (> F K))))
                    (prompt "\n  REPETIU! "))
             (if F (progn
                     (setq A1P (open A1 "r")
                           O (fix F) F (repeat O (read-line A1P)))
                     (PREFORM)
                     (FORMAT (list 'F2 'F3 'F5 'F6 'F8 'F9))
                     (IMPR F1 F2 F3 F4 F5 F6 F7 F8 F9)
                     (close A1P)))))
  (princ))
```

Acabarem el capítol amb una relació de les funcions principals, caracteritzades com a noves ordres d'AutoCAD (**C:FES-123456789**, **C:CARREGA-123456789**, **C:FES-TRIADES**, **C:FES-STRIADES**, **C:FES-SUDOKUS**, **C:FES-SSUDOKUS** i **C:VIS-XSN**, a les quals afegirem **LOCL1**, que l'usuari haurà de reclamar tot sovint per conèixer els valors que cal subministrar a algunes de les altres) i amb unes pautes sobre la seva utilització.

La primera es refereix a la necessitat de crear l'arxiu de files 123456789.txt amb **C:FES-123456789** i disposar de la definició de **C:CARREGA-123456789**, tant si teniu intenció de construir arxius XSN-... de solucions normalitzades i seleccionar-ne unes quantes amb l'ajut de **C:VIS-XSN**, aprofitant-les per a l'opció SOLUCIÓ COPIADA del programa SUDÒKULUM que desenvoluparem a partir del capítol següent (aprofitar-les, ara per ara, vol dir passar-les a paper i d'aquí transcriure-les manualment, tot i que seria fàcil d'instrumentar algun pont directe que, mitjançant una nova opció SOLUCIÓ CAPTURADA, permetés la transferència directa d'informació entre els dos sistemes), com si en aquest programa us estimeu més obtenir la SUDOKU-SOLUCIó

per altres mitjans però a l'hora de treure'n un SUDOKU-PROBLEMA voleu recórrer al mètode 2 (en el capítol 8 encara no té nom perquè no n'hi ha d'altre, en el 12 ja l'anomenem així perquè li ha sortit un competidor i en el 14 n'hi surten dos més).

Pel que fa a la resta de funcions principals, i sense excloure altres combinacions dictades pels interessos concrets de cada usuari, són dues les formes d'actuar que semblen respondre a les exigències més habituals:

- Si només interessa disposar del conjunt de solucions normalitzades definit per tres conjunts concrets de tríades (és a dir, per tres files concretes, que seran la 1ª, 4ª i 7ª en totes les solucions), el més recomanable serà utilitzar **C:FES-TRIADES** per construir els arxius KOMPAT-<pF1>, KOMPAT-<pF4> i KOMPAT-<pF7>, i després **C:FES-SUDOKUS** per construir l'arxiu XSN-<pF1>-<pF4>-<pF7>. I el que val per un conjunt també és aplicable a uns quants, sempre que les posicions <pF1> (o les <pF4>, o les <pF7>) no siguin consecutives, cas en què es podria recórrer a **C:FES-STRIADES** per obtenir d'una sola tirada els arxius ordenats KOMPAT-<pF>. Fins i tot, en el cas particular que les posicions <pF7> fossin les ordenades, podríem construir la sèrie d'arxius KOMPAT-<pF7> amb **C:FES-STRIADES** i, un cop disposéssim de tots els arxius KOMPAT-<pF1> i KOMPAT-<pF4> necessaris, cada sèrie d'arxius XSN-<pF1>-<pF4>-<pF7> corresponent a les posicions ordenades <pF7> podria ser obtinguda amb una execució de **C:FES-SSUDOKUS**.
- Si volem una sèrie ordenada de conjunts de solucions normalitzades, aleshores el mètode més senzill serà recórrer directament a **C:FES-SSUDOKUS**, que s'encarregarà de fabricar sobre la marxa els arxius KOMPAT-... que la construcció dels XSN-... ordenats vagi requerint (si n'hi hagués algun de fet prèviament, l'aprofitarà).

Tornem a començar: de la solució al problema

Començar a pensar en la presentació del programa (la interfície d'usuari) i en el procés d'extracció dels SUDOKUs-PROBLEMA d'una SUDOKU-SOLUCIÓ no va constituir de moment cap ruptura amb la tasca rutinària d'anar creant paral·lelament arxius de tríades normalitzades compatibles COMPAT-..., sinó una manera d'avançar feina, a l'espera que alguna idea feliç permetés d'accelerar la fabricació d'aquests arxius i abordar amb raonable eficiència la creació d'un macroarxiu únic o d'un conjunt d'arxius parcials de solucions normalitzades. Si finalment arribàvem a tenir el programa a punt però només una ínfima part de les solucions normalitzades, sempre podríem adduir que amb això ja n'hi havia prou per treballar (fer SUDOKUs-PROBLEMA i verificar la solidesa del programa), i que l'exhaustivitat del sistema només era una qüestió de temps. De fet, poca cosa més podíem fer en la línia del capítol 1, tret de seguir creant arxius COMPAT-... a estones lliures d'ordinador, perquè les poques coses que havíem pensat sobre la construcció de solucions normalitzades es basaven en la disponibilitat de tots aquests arxius (a diferència de l'orientació fragmentària de la revisió, basada a més en uns arxius KOMPAT-... molt més prims). Però de mica en mica aquests preparatius van anar cobrant vida pròpia, d'una banda per la desmotivació de l'autor en relació a una activitat que s'havia convertit en pura rutina, i de l'altra en adonar-se que la consecució d'un inventari universal, a més d'altament improbable, podia representar uns problemes de gestió insolubles.

En aquest procés d'emancipació i allunyament progressius dels propòsits inicials, podem considerar 3 etapes, que es van succeir obeint una lògica interna que havia deixat enrera l'horitzó inabastable de l'inventari de solucions, fins al punt que, quan casualment l'autor va tenir notícia de l'estimació feta per B. Felgenhauer i F. Jarvis sobre el nombre de SUDOKUs-SOLUCIÓ possibles (una vegada va plantejar-se aquest problema, però no va trigar a adonar-se que sobrepassava de molt els seus migrats coneixements), no va sentir cap commoció sinó més aviat cert alleujament: ja feia temps que havia deixat d'escalfar l'ordinador i omplir el disc amb arxius inútils, conscient que el nou camí emprés estaria orientat millor o pitjor però el duria a algun lloc. En farem un breu repàs, abans de centrar-nos en la versió de **C:SUDOKULUM** que adoptarem com a punt de partença de l'exposició, en tenir ja tota aquella infraestructura que, sense grans canvis, ens permetrà de progressar en els objectius complementaris de crear SUDOKUs-SOLUCIÓ a partir de zero (funció **OMPLE-SUDOKU**) i d'obtenir SUDOKUs-PROBLEMA a partir d'una SUDOKU-SOLUCIÓ (**FES PÚBLIC**).

El primer pas va consistir a dissenyar algunes eines per permutar les files d'una tríada i les tríades d'una solució, per passar d'una solució normalitzada a alguna de les 1.296 variants ja no normalitzades però que segueixen sent SUDOKUs-SOLUCIÓ. La idea encara assumia la hipòtesis de plena disponibilitat de totes les solucions normalitzades, tot i que el mecanisme de visualització i elecció encara estava per definir (ja hem comentat que tampoc no vèiem clar quina estructura de dades podria suportar millor l'inventari), però podíem suplir provisionalment aquesta carència copiant de qualsevol lloc alguna SUDOKU-SOLUCIÓ i normalitzant-la manualment. Tret d'aquesta llicència, el programa s'havia de comportar com si de debò disposéssim del recull complet de solucions normalitzades i, en conseqüència, ens oferia la possibilitat de permutar les files dins de cada tríada i de permutar les tríades, "desnormalitzant" la solució; d'aquesta manera (atenció, perquè aquest era el punt més discutible del plantejament) era posible aconseguir qualsevol SUDOKU-SOLUCIÓ. A l'àrea gràfica es visualitzarien ambdues solucions: a l'esquerra la normalitzada (l'original, teòricament) i a la dreta la variant obtinguda mitjançant permutació de files i/o tríades. Sobre fons negre, els valors de les caselles es veurien en gris (no ens volíem distreure jugant amb color i especulant sobre la plasticitat de una o altra combinació), i canviarien a blanc quan les activéssim (decidíssim que el contingut d'una casella havia de fer-se públic, és a dir, havia de formar part del SUDOKU-PROBLEMA). Tot i que, d'entrada, l'activació es produïa fent clic en les caselles de la representació dreta (el canvi a blanc s'estenia a les dues), podia passar-se d'una banda a l'altra. Però, quan encara estàvem ajustant aquest mecanisme i tot just havíem començat a pensar en un procediment per comptabilitzar les solucions compatibles amb les caselles públiques (procediment que esdevindria més simple limitant-nos a solucions normalitzades) i el moment en què iniciaríem aquest recompte, vam deixar en suspens aquestes feines perquè cada vegada era més gran la sensació d'haver arrencat en fals, amb allò de "qualsevol SUDOKU-SOLUCIÓ".

La segona va ser una breu etapa de transició en què vam reconsiderar l'orientació equívoca que pretenia fer creure a l'usuari que el programa estava posant al seu abast totes les SUDOKUs-SOLUCIÓ, per tal que en triés una com a punt de partença. Era veritat que, multiplicant amb 1.296 permutacions de files i/o tríades totes i cadascuna de las les solucions normalitzades, s'obtindria el conjunt de totes les SUDOKUs-SOLUCIÓ, però, fins i tot disposant de l'inventari complet de les primeres (que, òbviament no era el cas), era excessiu vendre el producte afirmant que a la pràctica això equivalia a disposar de l'inventari complet de les segones, perquè una cosa era escollir directament aquella SUDOKU-SOLUCIÓ que ens fes més patxoca (perquè les teníem totes a la vista) i una de ben diferent seleccionar una solució normalitzada i després permutar files i/o tríades per obtenir alguna de les 1.296 variants, procés aquest darrer en què ja era del tot inapropiat parlar d'elecció.

D'altra banda, ens havíem adonat que, amb no molt més esforç, es podia ampliar el repertori de transformacións d'una SUDOKU-SOLUCIÓ a la permutació de les columnes d'una tríada (de columnes, és clar), a la permutació de tríades de columnes, a la transposició de files i columnes, a les simetries respecte a la 5ª fila o a la 5ª columna i a la permutació dels valors 1... 9 de les caselles, manipulacions totes que podien encadenar-se i el resultat de les quals seguia sent una SUDOKU-SOLUCIÓ. Fins i tot vam veure que, en determinades circumstàncies, la permutació de files o de columnes de diferents tríades no malmetia la caracterització de SUDOKU-SOLUCIÓ. Tot això, com és lògic, retornant a la categoria general de solucions i oblidant-nos, de moment, dels requisits addicionals propis de les solucions normalitzades.

Aquesta ampliació de possibilitats ens va dur a un canvi de criteri: si no teníem accés directe a totes les SUDOKUs-SOLUCIÓ, i el més aproximat que podíem oferir-li a l'usuari era un inventari restringit a les solucions normalitzades (suposant que ens en sortíssim, d'aquesta empresa), que l'abocava a practicar permutacions per aproximar-se a la imatge preconcebuda de SUDOKU-SOLUCIÓ que pogués tenir (total o fragmentària), fet i fet era millor no obligar-lo a fer una tria entre elements, que no sempre seria immediat de relacionar amb aquesta imatge, prendre la decisió per ell i posar a la seva disposició una SUDOKU-SOLUCIÓ d'aspecte neutre i fàcil de manipular (en haver-hi una successió creixent, per exemple, a totes les files).

Però també a un miratge: enlluernats per aquesta sobtada obertura d'horitzons, i sense pensar-nos-ho dues vegades (si haguéssim reflexionat mínimament, no hauríem trigat a trobar algun contraexemple que desmentís la hipòtesi), vam a arribar a creure que "qualsevol SUDOKU-SOLUCIÓ pot obtenir-se de qualsevol altra SUDOKU-SOLUCIÓ mitjançant una seqüència limitada de permutacions com aquestes...", presentant les deu transformacions esmentades com a F1, F3, C1, C3, TR, FS, CS, NV, F2 i C2, i acabàvem afirmant que, si era indiferent la SUDOKU-SOLUCIÓ de partença, en proposaríem una força asèptica: la reproduïm a la dreta.

9 1 2	3 4 5	6 7 8
6 7 8	9 1 2	3 4 5
3 4 5	6 7 8	9 1 2
8 9 1	2 3 4	5 6 7
5 6 7	8 9 1	2 3 4
2 3 4	5 6 7	8 9 1
7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9

Potser també hi va ajudar que fos una manera prou airosa de sortir del cul-de-sac del capítol precedent, però cal reconèixer que ens vam ficar de peus a la galleda. El més curiós és que aquesta tremenda badada va ser el detonant que va consumar la ruptura *de facto* amb els plantejaments heretats del capítol 1 i, en concret, amb l'omnipresència de les solucions normalitzades, que ja només intervindrien com un recurs per simplificar el procediment de recompte de les solucions compatibles amb la part pública del sudoku, a partir d'un llindar d'ocupació que caldria establir.

No trigaríem a adonar-nos de la patinada i, no només el text adreçat a l'usuari es va canviar per "qualsevol SUDOKU-SOLUCIÓ pot obtenir-se de qualsevol altra SUDOKU-SOLUCIÓ mitjançant una seqüència limitada de permutacions com aquestes...", sinó que l'obligada acceptació de la SUDOKU-SOLUCIÓ reproduïda més amunt (ara batejada com "SOLUCIÓ CANÒNICA") passava a ser una més de les tres opcions que s'oferien. Efectivament, després de presentar les deu transformacions el text seguia així:

El programa permet d'aplicar aquestes transformacions a qualsevol SUDOKU-SOLUCIÓ i crear un SUDOKU-PROBLEMA a base de decidir quines caselles han de ser visibles per al jugador. El SUDOKU-SOLUCIÓ:

A) Es pot anar improvisant, amb l'ajut del programa.

B) Pot ser qualsevol solució donada, que caldrà transcriure casella a casella.

C) Si la solució es indiferent pot adoptarse aquesta, que anomenem Canónica:

Tot seguit es mostrava la SOLUCIÓ CANÒNICA i es sol·licitava a l'usuari que triés.

La presentació encara s'havia d'endreçar, mostrant de primer les opcions A, B i C, i no fent esment de les possibilitats de modificar la SUDOKU-SOLUCIÓ fins no haver efectuat l'elecció i tenir-la representada en pantalla gràfica; la tan desordenada exposició (afirmar que el propòsit del programa és l'obtenció d'un SUDOKU-PROBLEMA a partir d'una SUDOKU-SOLUCIÓ i que combinant 10 transformacions la podem canviar, sense que deixi de ser SUDOKU-SOLUCIÓ; després, mostrar les tres formes d'accedir-hi i finalment, tenint-ne ja una, permetre d'aplicar-li aquelles transformacions) només s'explica tenint en compte que procedíem d'una segona etapa en què havíem de partir obligatòriament d'una solució. Però ara ja ens trobàvem en l'etapa final, i estalviarem avenços i successives millores per tal de presentar (i justificar, si cal) la forma definitiva que va adoptar i que es mantindrà fins arribar al capítol 12 (*Un nou camí i també una drecera*). Abans d'iniciar aquesta presentació, de tota manera, convindria aclarir la qüestió del perquè ens vam embolicar amb una opció A que havia de permetre fabricar sobre la marxa SUDOKUs-SOLUCIÓ; perquè, si s'hagués tractat únicament d'esmenar l'error d'atribuir a qualsevol solució la possibilitat de generar totes les demés (quan per aplicació de F1, F3, C1, C3, TR i NV només hi ha 1.218.998.108.160 variants, com justificarem més endavant), era suficient crear l'opció B, mitjançant la qual hauríem pogut copiar SUDOKUs-SOLUCIÓ publicades (per exemple, per veure si activant altres caselles aconseguíem un SUDOKU-PROBLEMA més despoblat que aquell que se'ns proposava) o procedents de l'inventari de solucions (en aquesta improbable situació, ens hauríem de conformar amb les normalitzades). L'escepticisme sobre la viabilitat d'aquest inventari no va ser l'única raó, sinó el que hem dit a la pàgina precedent en relació a les motivacions de l'usuari: per aproximar-se a la imatge preconcebuda de SUDOKU-SOLUCIÓ (total o fragmentària) que pogués tenir. Què volíem dir amb això? Doncs que podia tenir el caprici d'explorar SUDOKUs-PROBLEMA alternatius, deduïbles d'una SUDOKU-SOLUCIÓ coneguda (per a això ja n'hi hauria prou amb l'opció B) o, inversament, de compondre una SUDOKU-SOLUCIÓ nova copiant algunes caselles d'una altra SUDOKU-SOLUCIÓ (part que podia coincidir o no amb el SUDOKU-PROBLEMA proposat) i emplenant la resta amb valors compatibles. Però també tenia dret a deixar-se endur per vel·leïtats esteticistes: per exemple, emplenant amb valors no aleatoris (successius, alternant parells i senars, etc.), caselles que formessin determinades figures geomètriques, com ara aquestes tres,

5 X X	X X X	X X **6**	X X X	X **5** X	X X X	X X X	**8** X **1**	X X X
X **4** X	X X X	X **5** X	X X X	**4** X **3**	X X X	X X **7**	X X X	**2** X X
X X **3**	X X X	**4** X X	X X **3**	X X X	**4** X X	X **6** X	X X X	X **3** X
X X X	**2** X **3**	X X X	X **2** X	X X X	X **5** X	**5** X X	X X X	X X **4**
X X X	X X X	X X X	**4** X X	X X X	X X **6**	X X X	X X X	X X X
X X X	**1** X **4**	X X X	X **3** X	X X X	X **4** X	**4** X X	X X X	X X **5**
X X **2**	X X X	**5** X X	X X **2**	X X X	**5** X X	X **3** X	X X X	X **6** X
X **3** X	X X X	X **6** X	X X X	**1** X **6**	X X X	X X **2**	X X X	**7** X X
4 X X	X X X	X X **7**	X X X	X **7** X	X X X	X X X	**1** X **8**	X X X

en què les X representen qualsevol valor (compatible amb els valors explícits) que convé introduir després (si aquests valors explícits -en negreta- no s'assignen de bon principi, potser després no ho podrem fer, per incompatibilitat amb la resta). Per a gairebé totes aquestes coses ens serà útil (si no imprescindible) l'opció A.

Si ara mateix i sense més comentaris entréssim en el moll d'aquest capítol, que és la descripció de la interfície gràfica i de la carcassa del programa (**C:SUDOKULUM** i funcions de suport a **OMPLE-SUDOKU** i **FES-PUBLIC**), després d'haver justificat que convenia donar entrada a una opció A, estaríem edulcorant la història, en suggerir que de seguida vam canviar les prioritats i ens hi vam posar a treballar (l'ordre alfabètic A-B-C i el dels capítols de l'ÍNDEX abonarien aquest supòsit), però les coses no van anar tan rodades. És cert que el nou enfoc ja partia d'unes premises més realistes que les del capítol precedent (no volíem tenir totes les solucions, sinó que enunciàvem tres formes de tenir-ne), però seguia tan desenfocat com abans perquè emfasitzava l'obtenció del SUDOKU-PROBLEMA a partir de la SUDOKU-SOLUCIÓ i no prestava massa atenció a l'obtenció d'aquesta. De fet, vam començar a treballar abans amb **FES-PUBLIC** (capítols *8- Detectar solucions compatibles* i *10- Solucions compatibles ho són totes les que hi són...*) que amb **OMPLE-SUDOKU** (capítol *3- Etapa prèvia: crear una solució, I*); dintre de **FES-PUBLIC**, abans de conèixer la tasca de Gordon Royle, **SUD-MIN** no es disparava fins arribar a 23 caselles plenes (capítol *9- Condicions mínimes*), i durant força temps vam creure que quan només quedava una solució normalitzada ja erem al cap del carrer (capítols *10- Solucions compatibles ho són totes les que hi són...* i *11- ... però no hi eren totes les que ho són*).

Feta aquesta precisió, s'entendrà millor que deixem per als capítols monogràfics corresponents l'estudi d'**OMPLE-SUDOKU** i de **FES-PUBLIC**, i que ara ens limitem a fer una exposició sumària del programa i a la presentació del seu entorn instrumental.

No cal executar **SUDOKULUM** just en iniciar un nou dibuix, sinó que la nova ordre pot ser invocada en qualsevol moment i sobre qualsevol dibuix, perquè després de completar-ne l'avaluació o d'interrompre-la podrem recuperar tot el que hi havia en pantalla amb un senzill **H** (o, el seu equivalent, **DESHACER 1**), tot i que en el segon cas caldrà fer abans **DESHACER F.** Després d'una benviguda bastant poca-solta, **PREPGRAF-9*9** ens permet de maximitzar la finestra de text, ajustar a 3 línies la part que apareix en mode gràfic, desactivar en l'àrea "Elementos de presentación" (del quadre de diàleg **OPCIONES** > **Visualización**) totes les caselles, tret de la 1ª (**Mostrar fichas Presentación i Modelo**), i triar entre tres grisos per aconseguir el contrast i lluminositat adequats; fet això, prepara l'entorn gràfic que aviat descriurem i que bàsicament consta de dos escaquers 9×9, iguals però independents, a dreta i esquerra. Després li toca el torn a **PREPTEXT**, que explica succintament l'estratègia del programa, consistent a extreure un SUDOKU-PROBLEMA de la SUDOKU-SOLUCIÓ, i ens dóna a escollir entre tres maneres d'obtenir la segona: una SOLUCIÓ CREADA (opció A) o COPIADA (B), o la CANÒNICA (C), mostrada a la penúltima pàgina.

En principi, la solució triada (i aconseguida amb un mínim d'esforç però una mica més de paciència, en els casos A i B) apareix duplicada a dreta i esquerra, amb la retícula en gris (menys el contorn dels requadres 3×3, que és blanc) i els valors de les caselles en blanc, tot sobre fons negre. Però la funció **VARIACIONS** ens dóna la possibilitat de convertir-la en una altra SUDOKU-SOLUCIÓ (parlem de triar entre 1.218.998.108.160 variants, xifra ampliable a 33.022.859.292.057.600 en cert cas), mitjançant l'aplicació d'algunes transformacions elementals. Ens ho presenta així:

D'aquesta SUDOKU-SOLUCIÓ se'n poden treure d'altres, per aplicació reiterada de les transformacions següents:

F1: Permutació de Files d'una mateixa tríada
(6 permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).

F2: Permutació de Files de tríades diferents
(fins a 216 permutacions, però no sempre és possible).

F3: Permutació de tríades de Files (6 permutacions).

FS: Simetria respecte a la Fila central (es pot fer amb tres F1 i una F3).

C1: Permutació de Columnes d'una mateixa tríada
(6 permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).

C2: Permutació de Columnes de tríades diferents
(fins a 216 permutacions, però no sempre és possible).

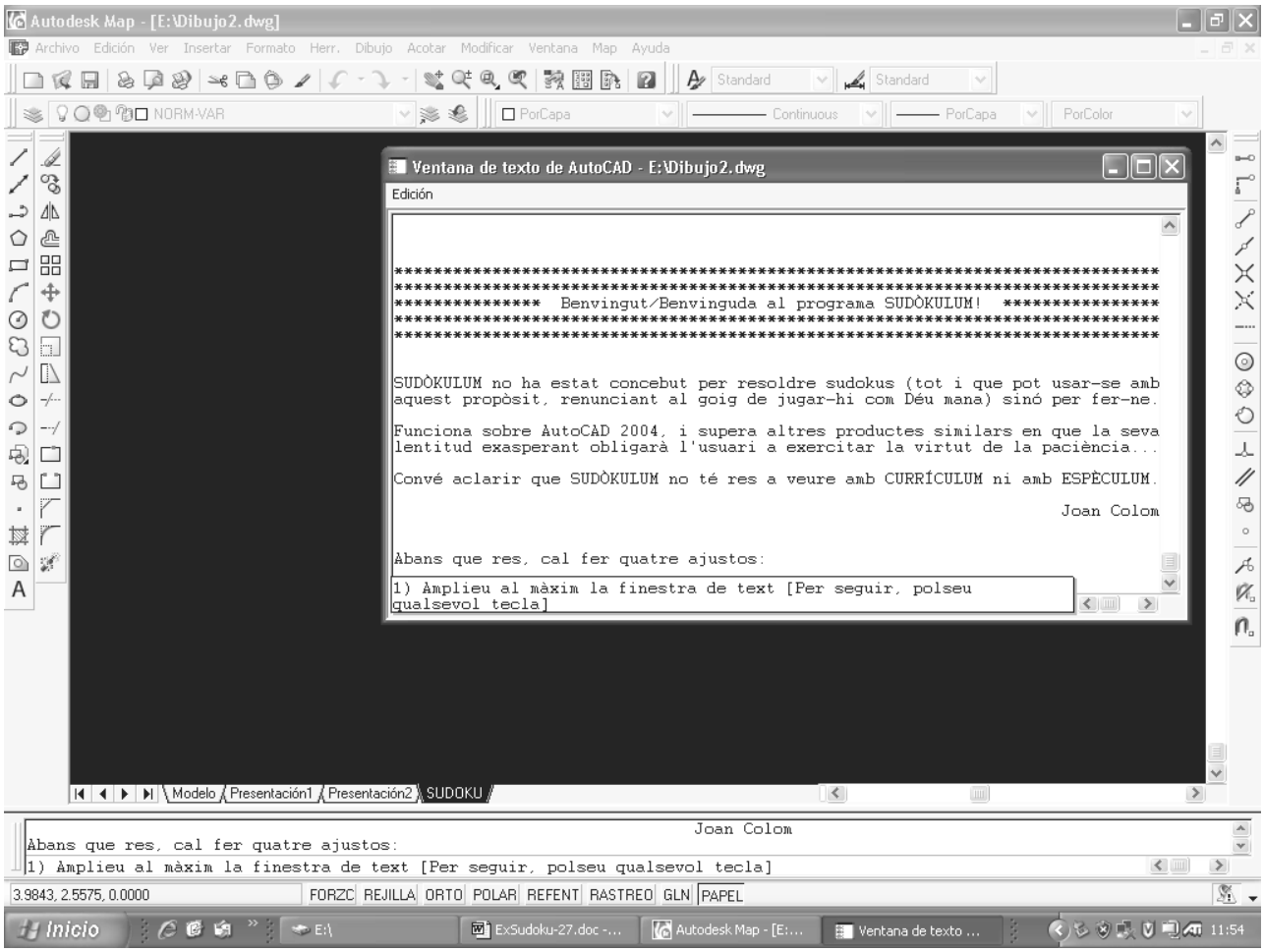
C3: Permutació de tríades de Columnes (6 permutacions).

CS: Simetria respecte a la Columna central (es pot fer amb tres C1 i una C3).

TR: Transposició de files i columnes.

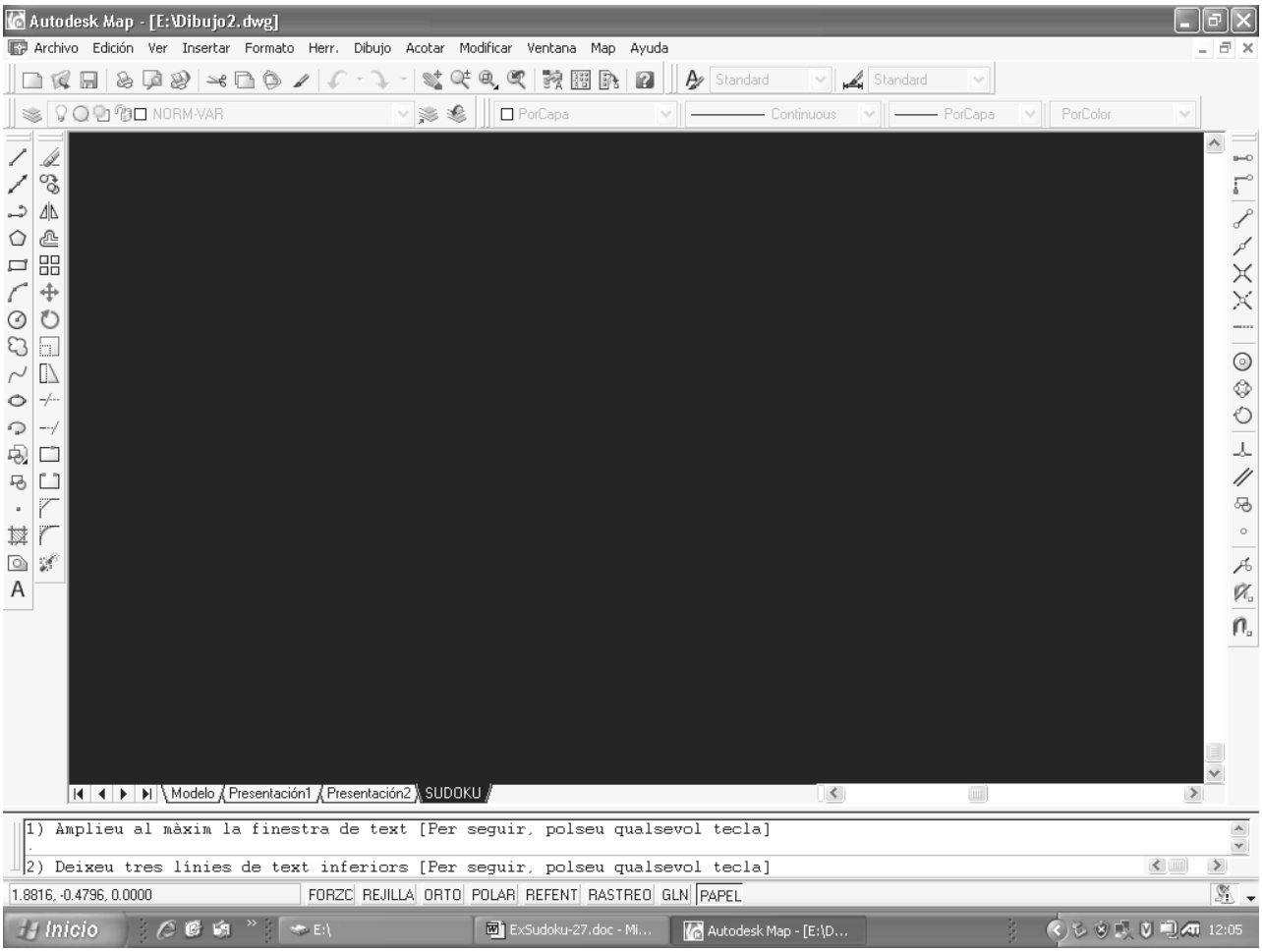
PV: Permutació entre Valors 1 a 9 estesa a tot el SUDOKU (tot i que muntat com una seqüència de permutacions binàries, hi ha 9! = 362.880 possibilitats).

(Obviem argumentar per què les transformacions F1, F3, FS -combinació de F1 i F3-, C1, C3, CS -combinació de C1 i C3-, TR i PV no afecten les condicions definitòries d'una SUDOKU-SOLUCIÓ; quant a F2 i F2, més endavant veurem en quins casos tampoc.) L'oferiment es va reiterant (la SOLUCIÓ CREADA, COPIADA o CANÒNICA es manté sobre l'escaquer 9×9 de l'esquerra, mentre que sobre el de la dreta veiem aparèixer els resultats de les successives transformacions, amb el peu VARIANT ESCOLLIDA), fins que diem prou. Cal dir que, arribats en aquest punt, el programa ens sol·licita la conformitat en relació a la VARIANT ESCOLLIDA i, si no li donem, torna a l'inici. Acceptades ja, la SOLUCIÓ CREADA, COPIADA o CANÒNICA queda emmagatzemada a ****9*9**** i la VARIANT ESCOLLIDA a ****9**9****: són llistes de 9 subllistes de 9 enters 1... 9 (tot això s'esdevé a la funció **CANON->VARIANT**, i aviat ho veureu amb més detall).

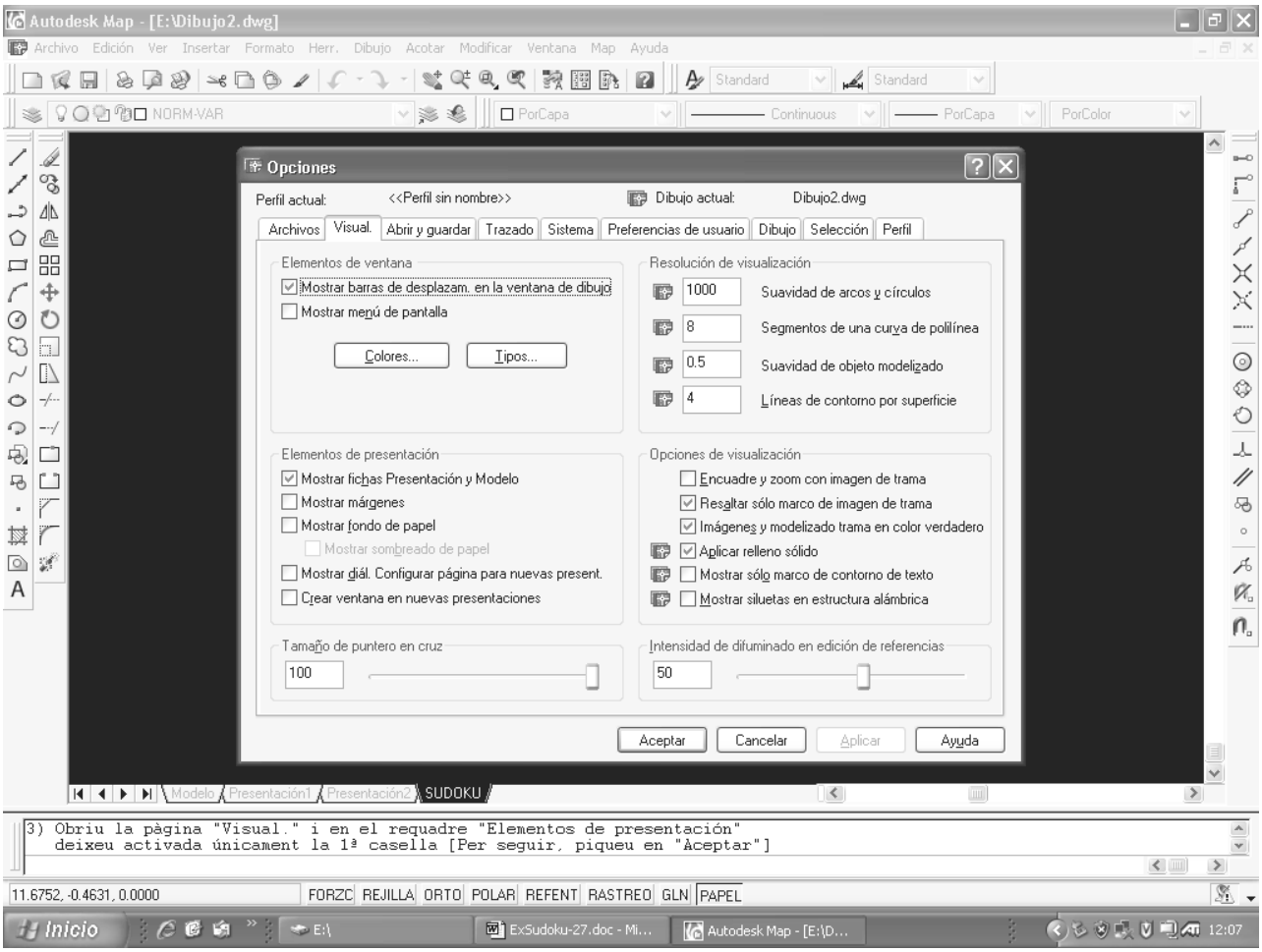


Per llegir bé la presentació cal maximitzar la finestra de text, Aquest ajust...

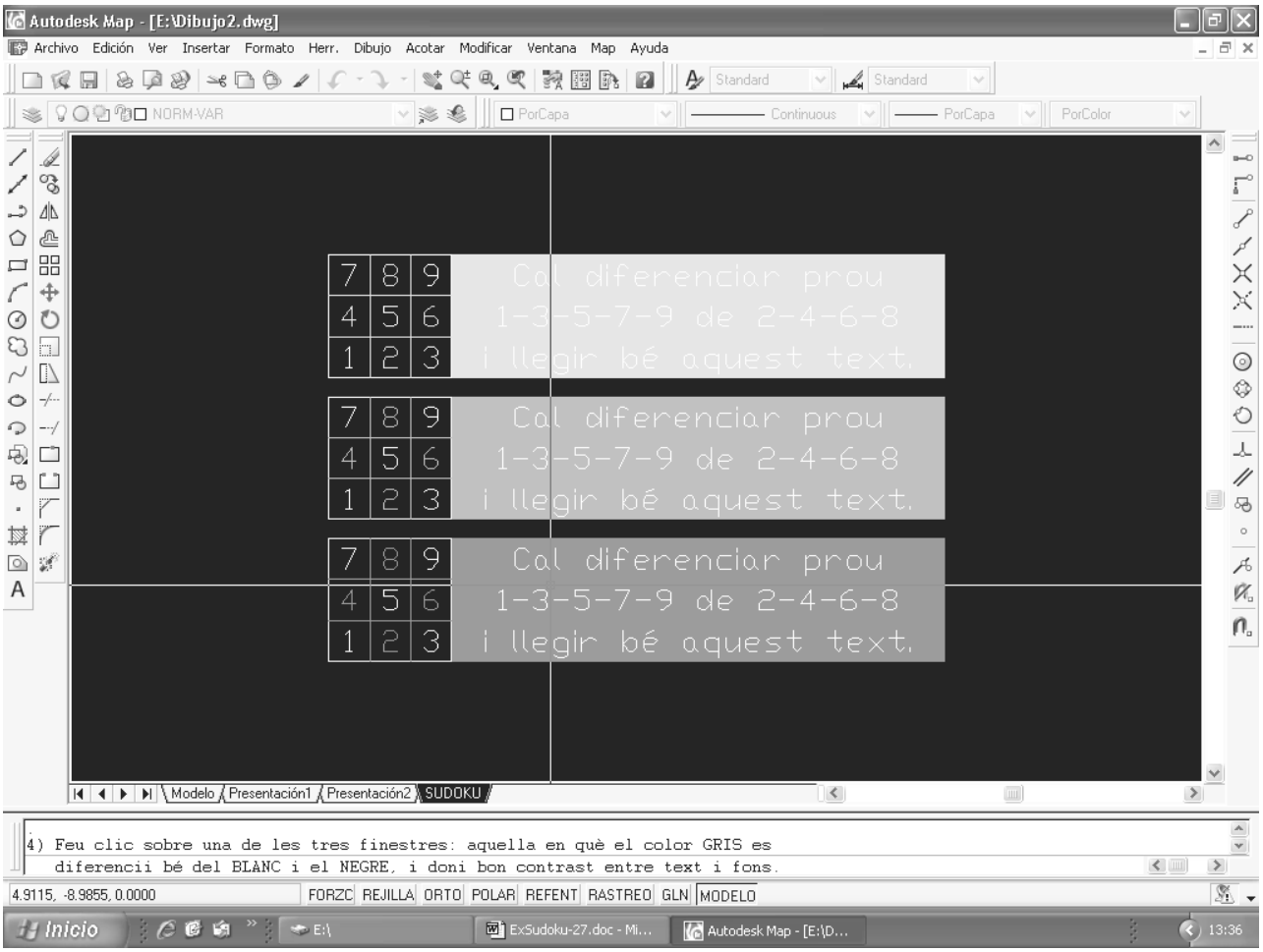
i tres més permetran que la visibilitat de textos i gràfics siguin els correctes.



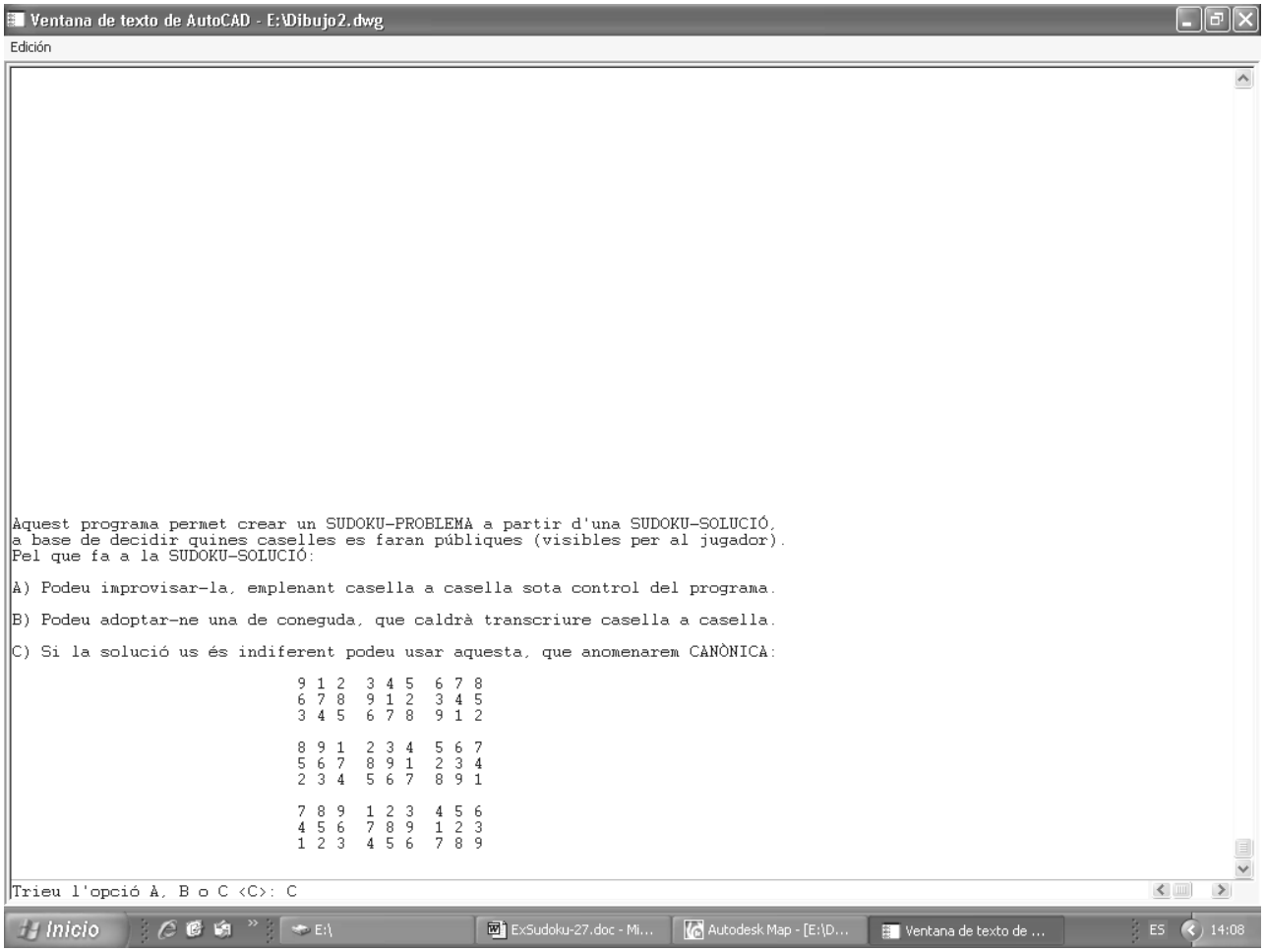
L'ajust següent consisteix a deixar l'àrea d'ordres inferior en 3 línies de text.



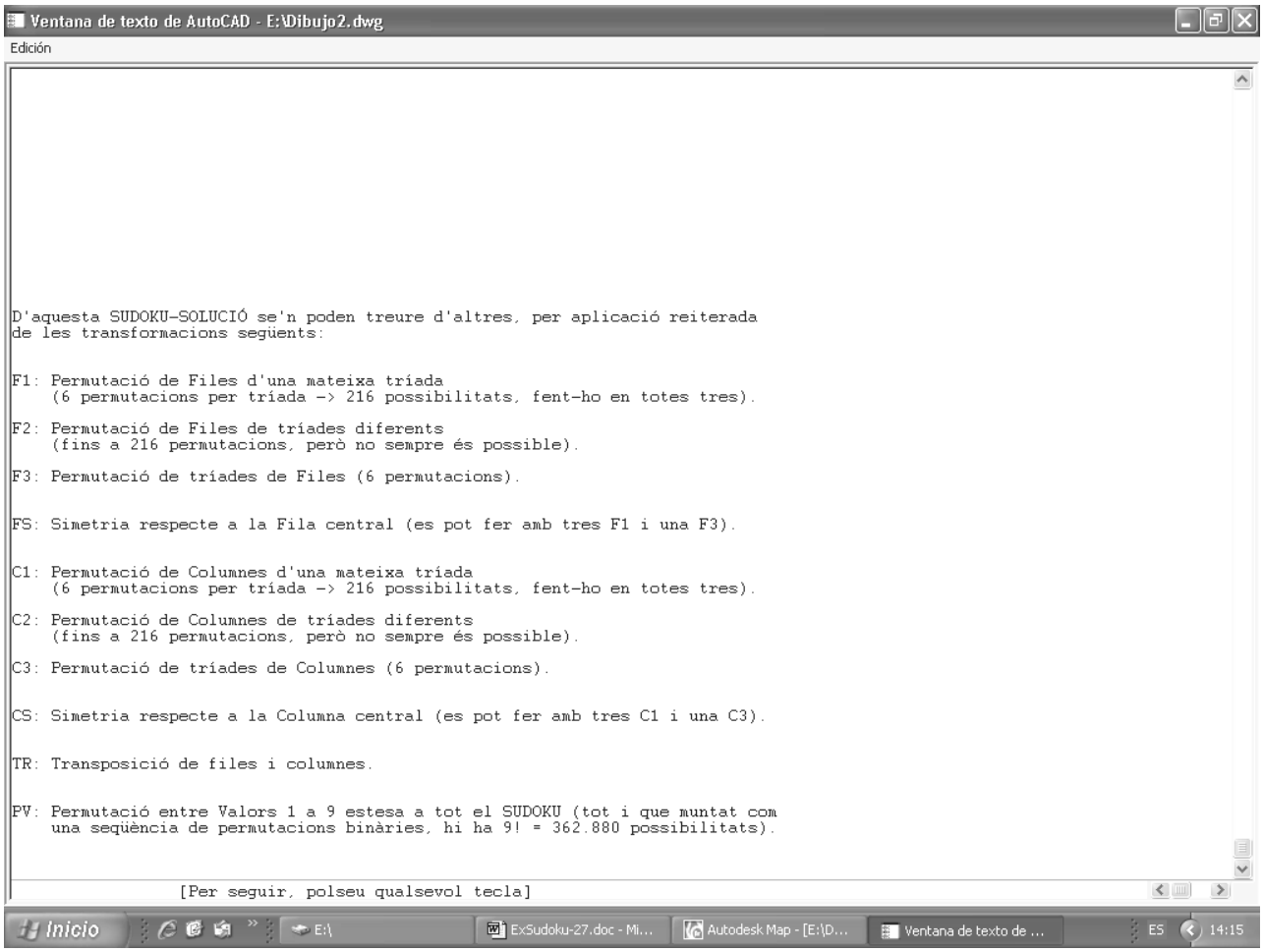
En **Opciones > Visual. > Elementos de presentación,** només la 1ª opció activada.



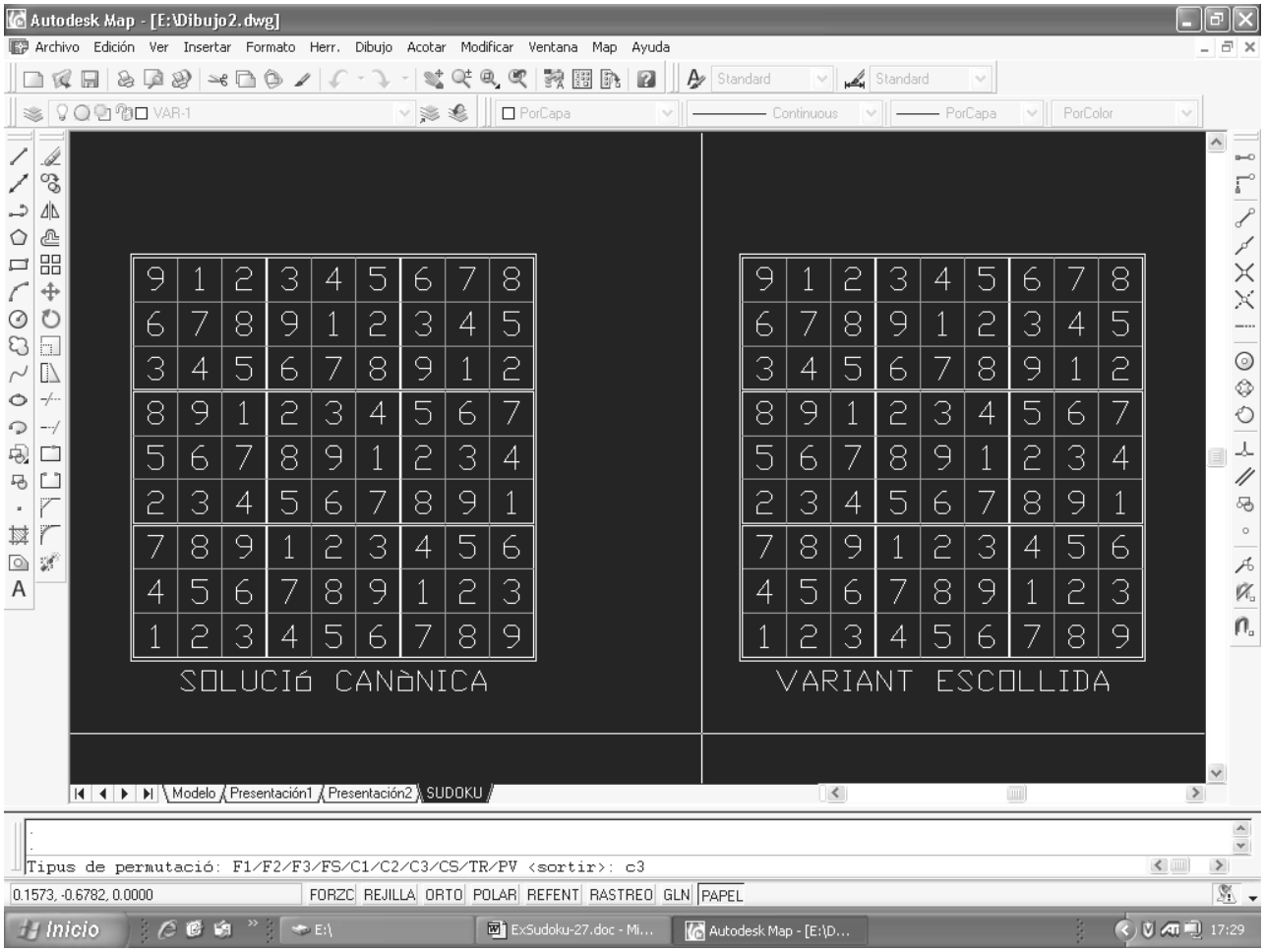
Cal escollir un gris per a color de text (fons negre) i com a fons (text blanc).



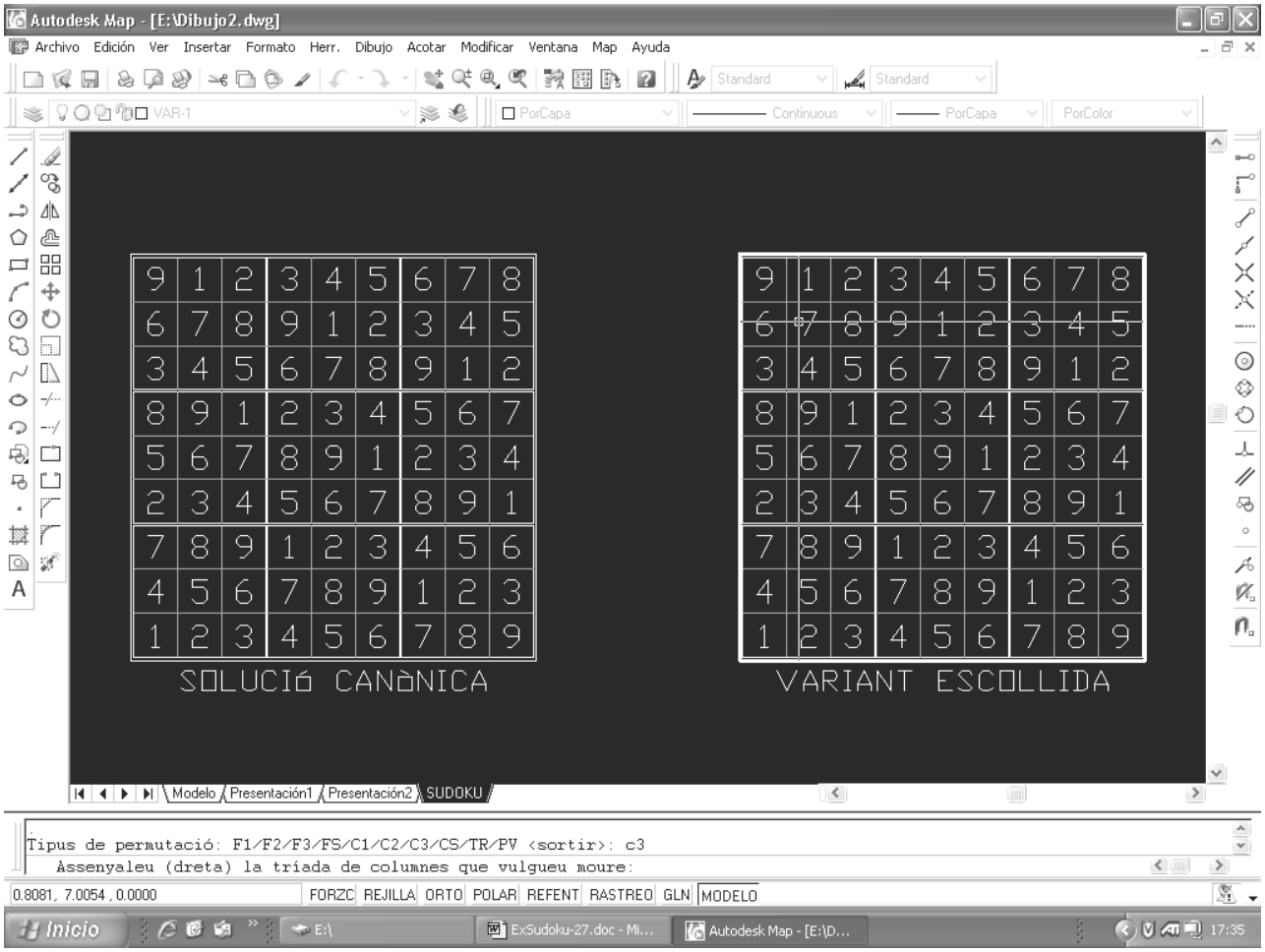
Entre crear una SUDOKU-SOLUCIÓ (A), copiar-la (B) o adoptar la mostrada (C)...



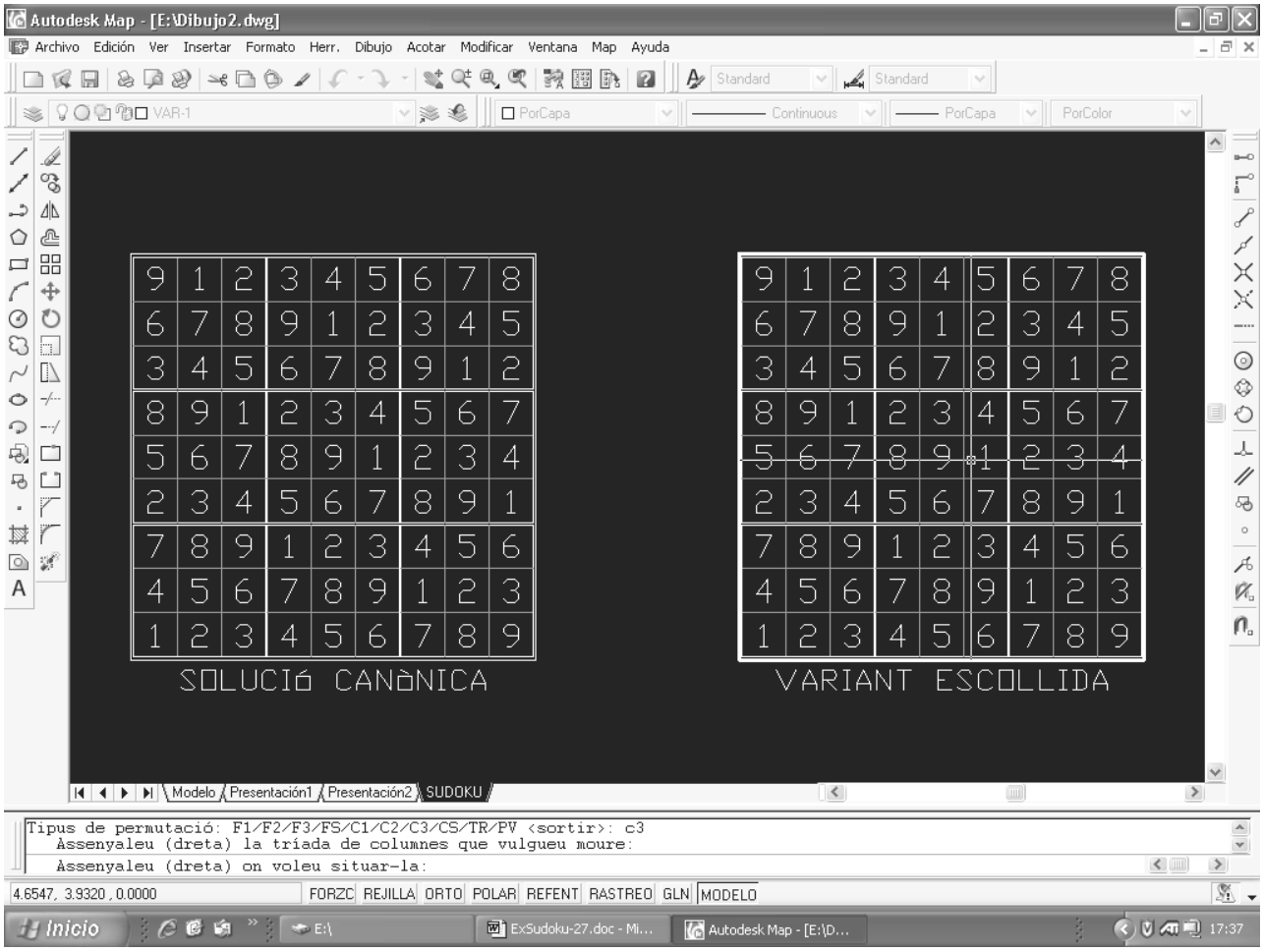
s'ha preferit l'opció C, que mutarem aplicant-li tres d'aquestes transformacions.



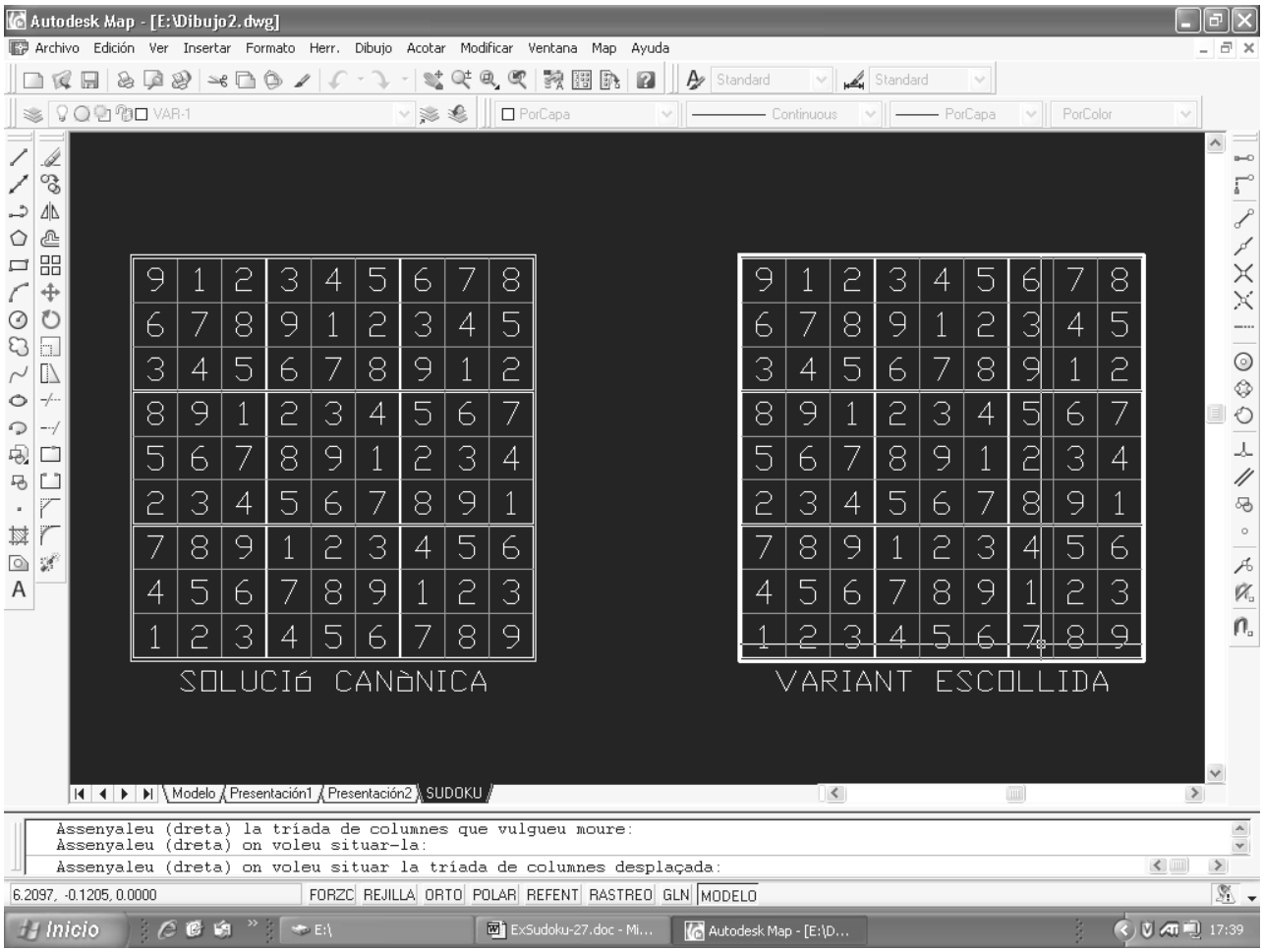
De primer, permutarem les posicions dels tres requadres 3×9.



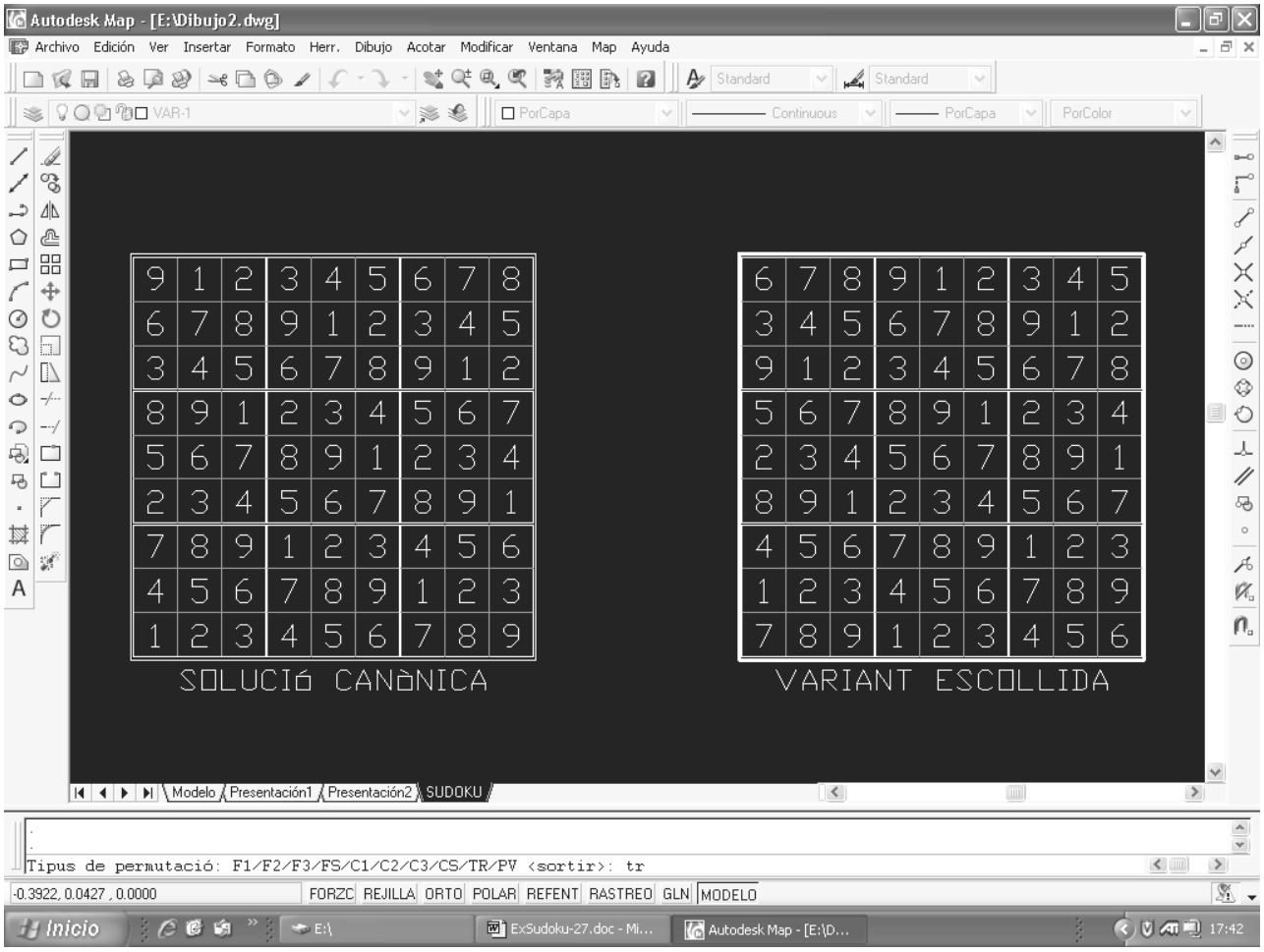
Cal precisar que el primer requadre 3×9 (clic)...



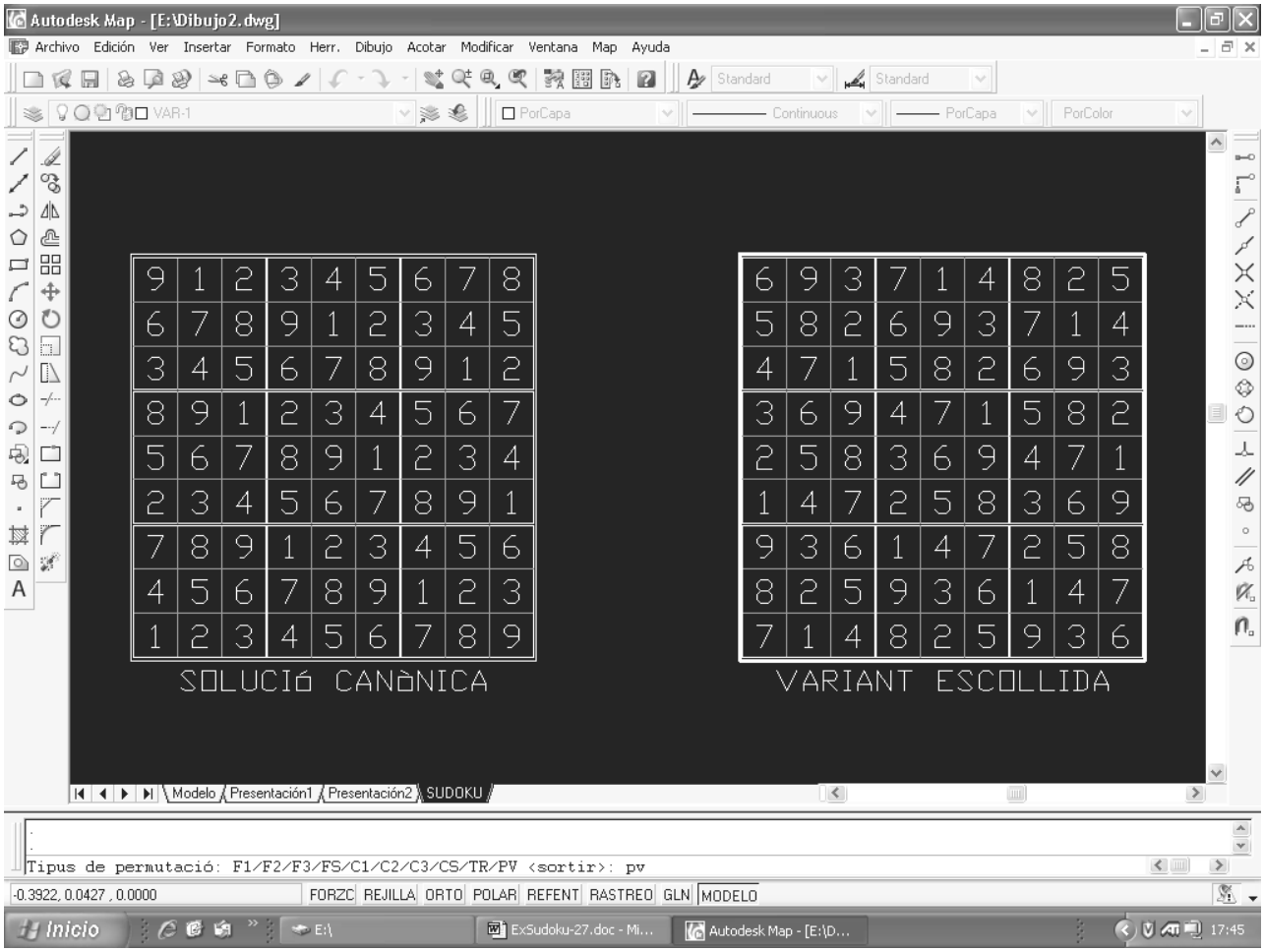
volem col·locar-lo en segona posició (clic), i que el segon...



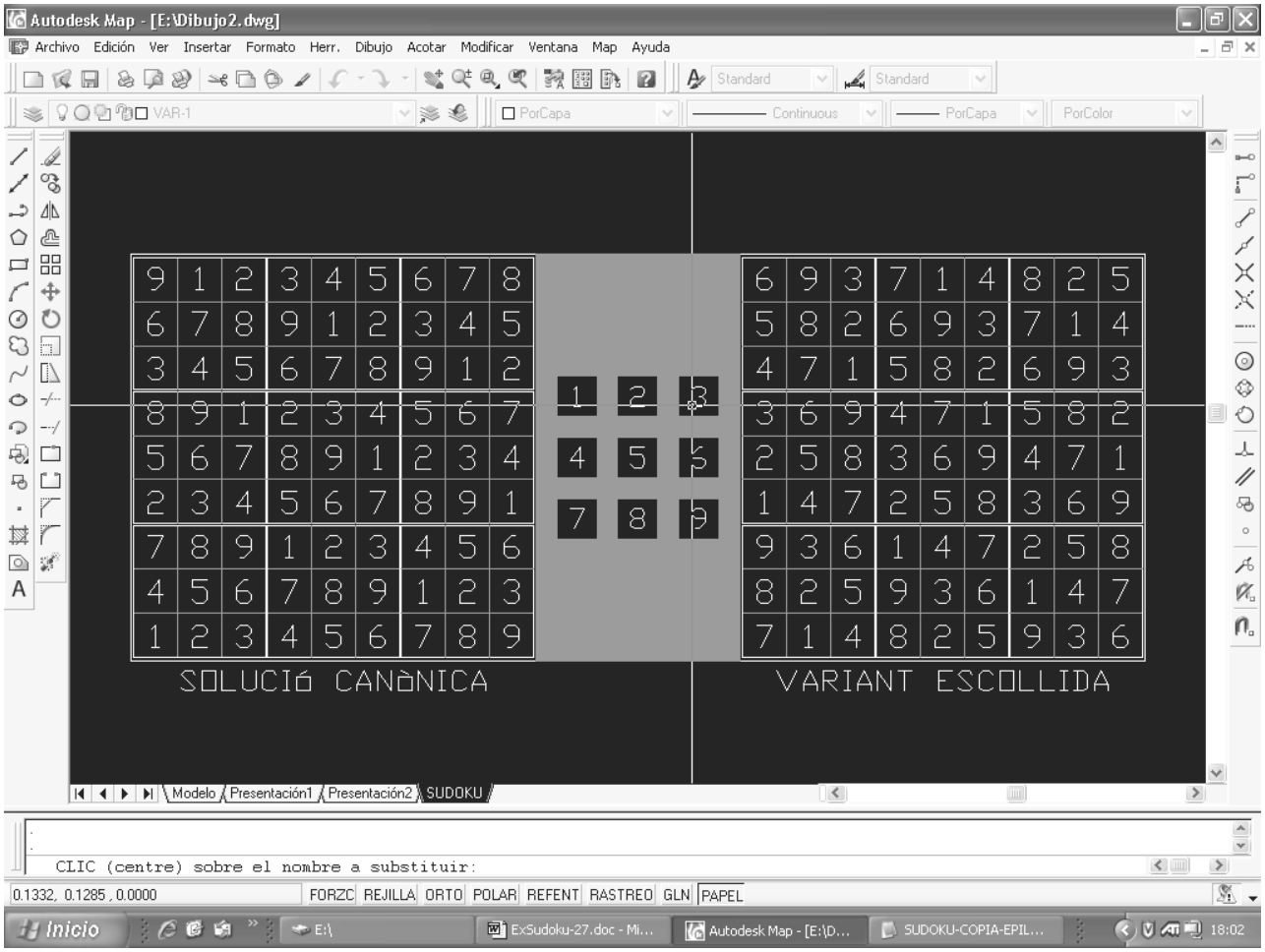
volem colocar-lo en tercera posició (clic). Òbviament, el tercer passarà...



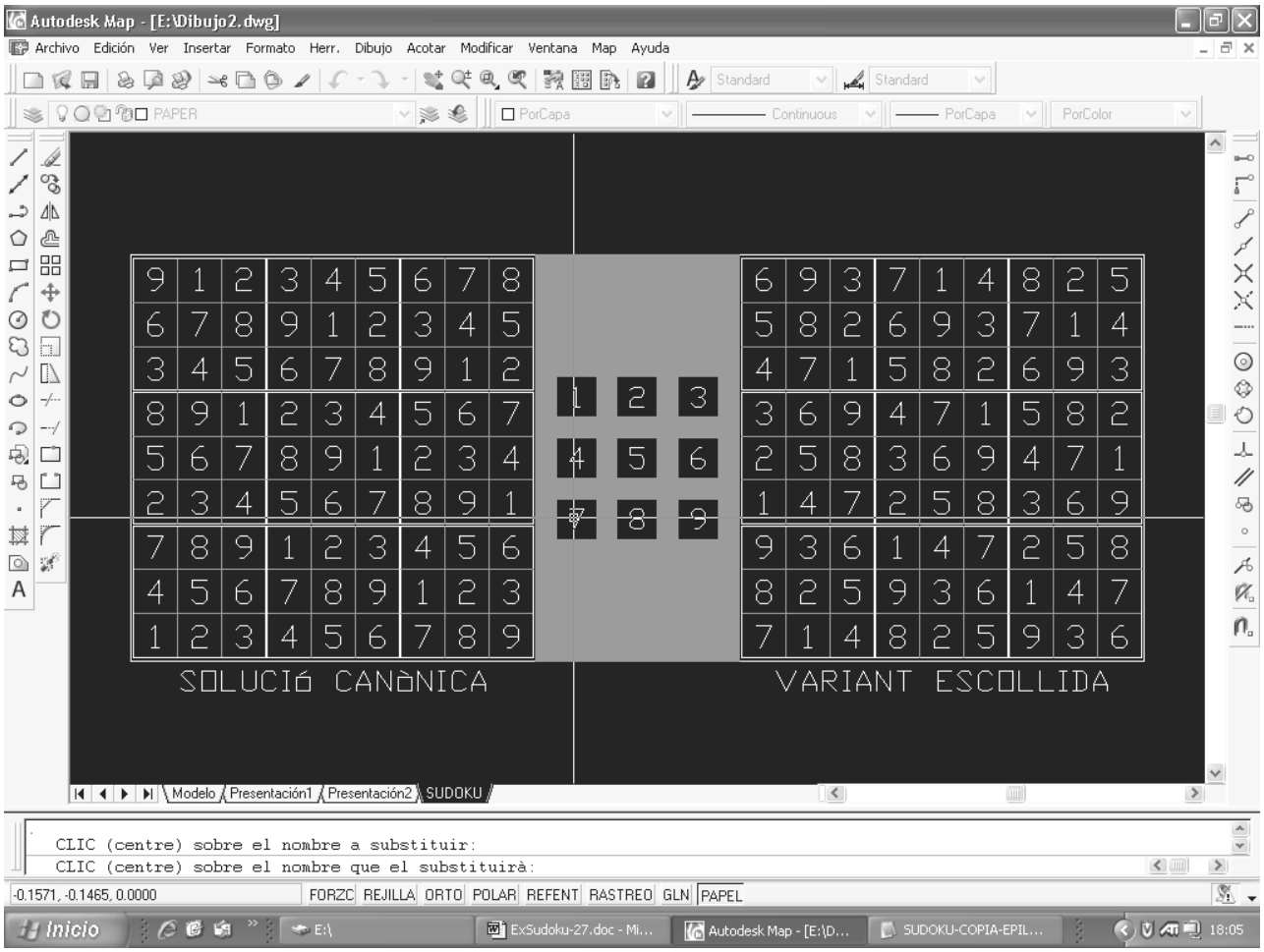
a primera posició. Seguirem, aplicant a VARIANT ESCOLLIDA una transposició...



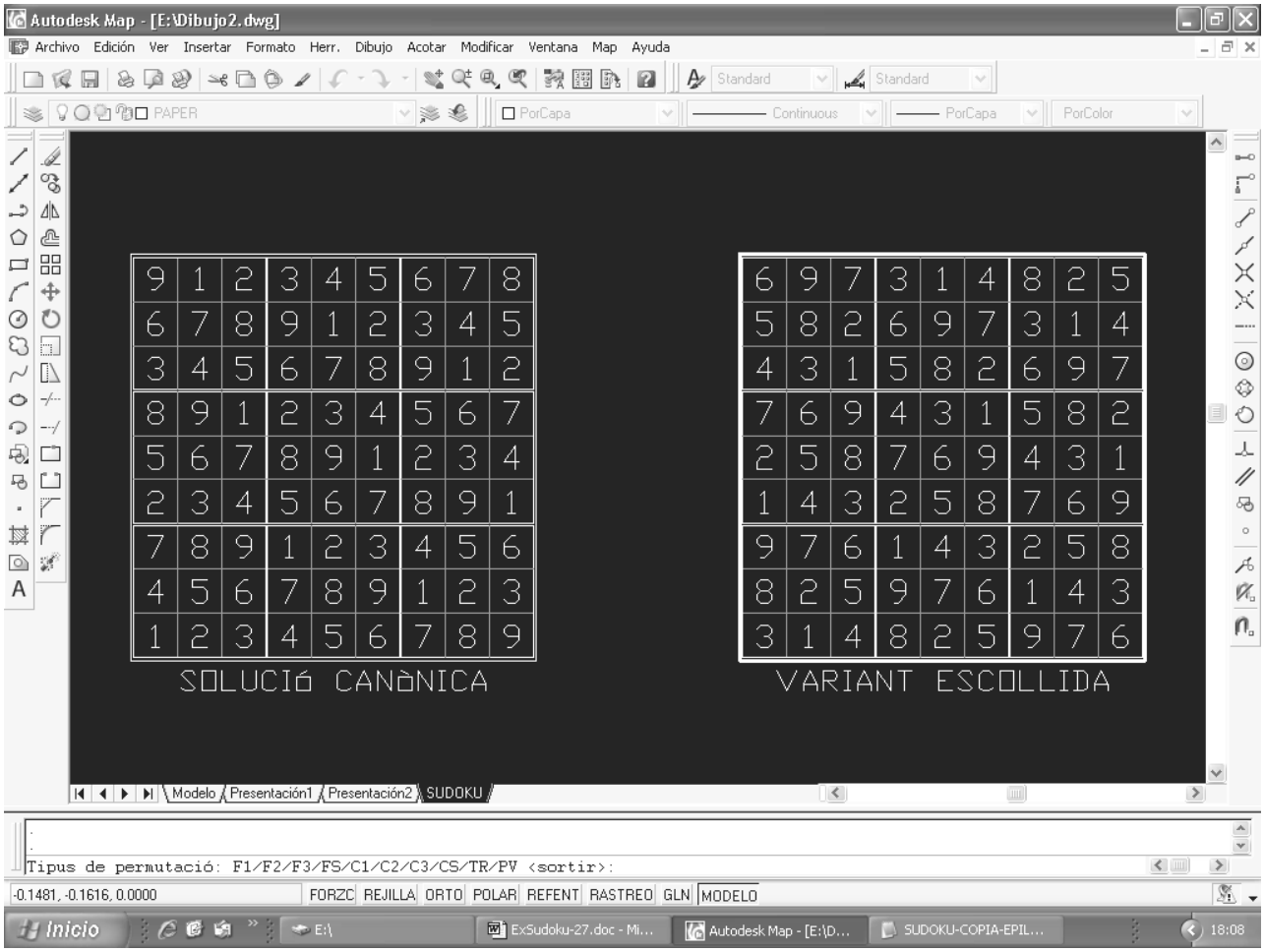
(intercanvi de files per columnes). Per acabar, aplicarem a l'últim resultat...



una permutació binària del contingut d'algunes caselles: les de valor **3 (clic)**...



per les de valor **7** (clic).



Sobre aquesta VARIANT ESCOLLIDA cercarem un SUDOKU-PROBLEMA (segueix en p. 361).

Llavors és quan SUDÒKULUM passa a la segona part: obtenir un SUDOKU-PROBLEMA de la SUDOKU-SOLUCIÓ finalment adoptada, és a dir, de la VARIANT ESCOLLIDA. Prèviament a les instruccions de funcionament d'aquest mòdul, se'ns informa que, treballant amb solucions normalitzades, el procés serà més ràpid:

Si anomenem SOLUCIÓ NORMALITZADA aquella en què les files de cada tríada estan ordenades numèricament d'avall cap amunt, i les tríades de files també estan ordenades atenent la fila inferior, a la finestra de l'esquerra veureu la V. E. NORMALITZADA, referida a la VARIANT ESCOLLIDA que resta a la dreta (per obtenir-la el programa haurà fet fins a tres permutacions F1 i una F3).

Jugant amb les solucions normalitzades serà força més ràpid el processat per detectar amb quantes caselles plenes la SUDOKU-SOLUCIÓ ja queda determinada.

Acceptat això, veurem com la SOLUCIÓ CREADA, COPIADA o CANÒNICA de l'esquerra és substituïda per una versió normalitzada de la VARIANT ESCOLLIDA de la dreta, canvi que inclou el del peu de la figura (que passa a anomenar-se V. E. NORMALITZADA) i que no fa sinó traduir gràficament l'actualització interna que ha sofert ****9*9**** (tot això s'esdevé a la funció **NORMTEXT-9*9**, i aviat en parlarem amb més detall). També se'ns recorda que l'arxiu de línies de SUDOKUs-SOLUCIÓ, 123456789.txt, s'ha d'haver creat (veieu **C:FES-123456789**, en el capítol precedent) i ha de figurar en algun directori de la ruta de cerca d'AutoCAD (és recomanable que se situï en la mateixa carpeta que l'arxiu de dibuix), per tal de carregar-lo des de **FES-PUBLIC:**

Cal tenir l'arxiu 123456789.txt a la ruta de cerca d'AutoCAD. Si no l'hi teniu, podeu copiar-lo en el directori d'aquest dibuix i després seguir com si no res.

Finalment, SUDÒKULUM ens proposa les regles del joc següents (funció **EXPLICACIÓ**):

Per construir el SUDOKU-PROBLEMA a partir d'aquesta SUDOKU-SOLUCIÓ, haureu de seguir els passos següents:

1) Com que els valors quedaran velats (en gris), haureu d'anar fent clic a les caselles que vulgueu convertir en públiques (visibles per al jugador). Els clics poden fer-se sobre la V. E. NORMALITZADA o sobre la VARIANT ESCOLLIDA (esquerra o dreta) però per passar de l'una a l'altra caldrà un altre clic.

2) Les caselles públiques quedaran destacades (blanc/negre) en ambdues bandes, però, si hi torneu a fer un clic, passaran de nou a ser secretes (en gris).

3) Quan el conjunt de caselles públiques assoleixi les condicions necessàries per a una solució única, es produirà una llarga pausa, durant la qual només el text inferior revelarà activitat. Quan aquesta s'acabi, un informe dirà quantes SOLUCIONS NORMALITZADES són compatibles amb la V. E. NORMALITZADA.

4) Si n'hi ha més d'una, haureu de seguir fent clic fins a tenir-ne només una.

Ara ens trobem un altre cop amb els dos escaquers 9×9, però el contingut de les caselles és gris (un privilegi que només se'ns concedeix a nosaltres, autors del SUDOKU-PROBLEMA que n'ha de sortir, perquè el seu destinatari ja no tindrà accés a aquesta informació), esperant repintar en blanc les que li ordenem, que seran guardades en les pseudomatrius ****V*V**** i ****V**V**** (esquerra i dreta), anàlogues a ****9*9**** i ****9**9**** amb una excepció: els elements de les 9 subllistes passen a ser enters 1... 9 en activar-se la casella corresponent, però mentrestant representen la seva posició matricial (coordenades discretes). D'entrada la finestra actual és la dreta (VARIANT ESCOLLIDA), però en qualsevol moment podem passar a l'esquerra (V. E. NORMALITZADA), fer-hi clic perquè sigui l'actual i seguir activant caselles des d'aquí. També podem rectificar, fent clic de nou sobre les caselles activades.

I ho deixem aquí, a les portes d'una descripció en profunditat que encetarem en el capítol 8 (construcció de **FES-PUBLIC**), de forma anàloga a l'el·lipsi d'abans sobre l'opció A per per no envair competències del següent (construcció d'**OMPLE-SUDOKU**). A canvi d'això, i abans de tractar amb cert detall del muntatge que s'ocupa de les transformacions SOLUCIÓ CREADA, COPIADA o CANÒNICA → VARIANT ESCOLLIDA i VARIANT ESCOLLIDA → V. E. NORMALITZADA, ara sí que convé referir-nos a l'entorn gràfic, en què s'ha procurat que el contingut de l'Espai Paper quedés restringit a la capa PAPER, i que el de l'Espai Model es distribuís entre les capes NORM-VAR, NORM-0,

NORM-1, VAR-0 i VAR-1. Després dels ajustos inicials, que ja hem comentat, i de la creació d'aquestes capes, la funció **PREPGRAF-9*9** definirà la fitxa de presentació SUDOKU, on s'obren dues finestres a esquerra i dreta, ajustades a l'escaquer 9×9 únic dibuixat en l'Espai Model. Això no es contradiu amb el que havíem manifestat, parlant de dos escaquers independents, sinó que l'efecte és precisament el descrit i s'aconsegueix aplicant un criteri d'economia de recursos: la quadrícula se situa a la capa NORM-VAR, bloquejada (per evitar que aquesta quadrícula resulti afectada per les nombroses ordres d'edició) i visible des d'ambdues finestres; els valors enters emmagatzemats en ****9*9**** o ****V*V**** es dibuixaran a la capa NORM-0 (gris) o a la NORM-1 (blanc), només visibles des de la finestra esquerra; els emmagatzemats en ****9**9**** o ****V**V**** es dibuixaran a la capa VAR-0 (gris) o a la VAR-1 (blanc), només visibles des de la finestra dreta. Perfecte, però, a banda d'obrir-hi dues finestres quadrades per veure-hi sengles escaquers 9×9 virtuals, ¿l'Espai Paper no l'aprofitarem per a res més? És clar que sí, perquè la capa PAPER serà el suport de tota una sèrie d'elements auxiliars:

- Al peu de cada finestra:
  - Els rètols SOLUCIÓ CREADA, COPIADA o CANÒNICA, V. E. NORMALITZADA (esquerra) i VARIANT ESCOLLIDA (dreta).

- En la franja compresa entre les dues finestres, que en la majoria de situacions pren l'aspecte de fons gris, sobre el qual apareixeran:
  - En fase de "presolució", amb opcions A (SOLUCIÓ CREADA) i B (SOLUCIÓ COPIADA), un caseller 3×3 i (dintre de les caselles) els valors 1... 9, sobre els quals l'usuari ha de fer clic per assignar-ne un a la casella actual de l'escaquer 9×9 de l'esquerra.
  - En fase de "postsolució", un símbol del Tao (o disc *taijitu*), que girarà cada vegada que moguem el cursor en la finestra actual (dreta o esquerra) i actuarà per indicar-nos que podem fer un clic sobre qualsevol casella, per activar-la o desactivar-la. En les primeres versions aquesta icona n'és l'únic indicador, però en l'últim capítol (*Consells pràctics per evitar una executio precox*) ja quedarà relegat a reforç o complement visual d'un missatge que, ocupant dues de les tres línies de l'àrea de text situada a la part inferior de la pantalla (sota l'àrea gràfica), serà més explícit i ens recordarà que *Podeu fer clic a qualsevol casella, sobre la finestra dreta o sobre l'esquerra (es passa d'una a l'altra amb un clic previ)*.
  - En fase de presolució amb l'opció A (mentre **OMPLE-SUDOKU** selecciona els valors 1... 9 que podrem assignar a la casella actual) o en la de postsolució (mentre **FES-PUBLIC** fa el recompte de solucions compatibles), uns rètols sobre el fons gris, que ens recorden que cal esperar la fi del procés per seguir fent clics.

- Sobre la finestra esquerra:
  - En fase de presolució amb l'opció B, una mira quadrada, que s'anirà desplaçant de casella en casella (d'esquerra a dreta i de dalt a baix) per facilitar-li a l'usuari la feina, dispensant-lo d'assenyalar les caselles que hagi d'activar.

Pel que fa al codi, descrit el funcionament de **C:SUDOKULUM** i havent deixat per als capítols següents l'estudi detallat d'**OMPLE-SUDOKU** i **FES-PUBLIC**, convé justificar **CANON->VARIANT** (amb les seves funcions subordinades), com a encarregada d'obtenir alguna VARIANT ESCOLLIDA de la SOLUCIÓ CREADA, COPIADA o CANÒNICA, i **NORMTEXT-9*9** (també amb les seves), com a encarregada de normalitzar aquesta VARIANT ESCOLLIDA (obtenint-ne la V. E. NORMALITZADA). I, per no perdre més temps, despatxarem aquí mateix tres funcions auxiliars compartides i usades directament per **C:SUDOKULUM**:
- **SEGUIR** representa una pausa en l'execució, que proseguirem amb qualsevol tecla.
- **RETOL-1** escriu un text **R** centrat en la posició **P**, sempre a l'Espai Paper. En el cas que **R** comenci amb "V." (de moment, només "V. E. NORMALITZADA") esborrarà el text que ocupi aquesta posició.
- **GRAF-9*9** esborra el contingut de les dues finestres (llevat de la quadrícula), emplenant la de l'esquerra amb els valors de ****9*9**** i la de la dreta amb els valors de ****9**9****.

La funció **CANON->VARIANT**, que (recordem-ho) s'executa immediatament després que **VARIACIONS** ens hagi assabentat de les 10 transformacions disponibles i de la forma de reclamar-les, ens convida repetidament a fer-ne ús, especificar-ne l'aplicació (si s'escau) i veure el resultat a la finestra dreta, mentre no donem una resposta nul·la (*sortir*): tan aviat podrem encadenar-ne una munió com no fer-ne servir cap. La millor manera d'estudiar-la serà examinar totes les respostes **RESP** possibles, i començarem per **"F1"**, **"F3"**, **"C1"** i **"C3"**, que són les que pertoquen a la funció **F/C**.

F/C començarà invocant **PREVIES** (una simple rutina de composició de textos per als missatges d'assistència) i **AREES**, funció que emmagatzema a la llista **GG** les zones rectangulars on l'usuari pot fer clic per designar files i columnes (opcions F1 i C1, i en aquests casos tindrem 9 zones), o tríades de files i tríades de columnes (opcions F3 i C3, i en aquests casos tindrem 3 zones). Aquests rectangles estaran representats pels vèrtexs de mínimes i màximes coordenades, però abans de seguir serà obligat fer un incís per parlar de la tolerància gràfica **TG** adoptada aquí i en altres funcions de SUDÒKULUM, que és **(0.48 0.48)**: quan l'usuari vulgui fer clic en una casella el centre de la qual és **P**, admetrem que no tingui massa punteria, sempre que el clic trepitgi un quadrat amb centre a **P** i de costat **2 × 0,48 = 0,96**; els vèrtexs inferior-esquerre i superior-dret d'aquest quadrat sortirien de les expressions **(mapcar '- P TG)** i **(mapcar '+ P TG)**, respectivament. Però no s'acabat l'incís, perquè algun lector que hagués executat **PREPGRAF-9*9**, si més no, podria objectar, carregat de raó, que l'àrea quadrada que visualitzen les finestres és de **9,00 × 9,00** Uds. d'Espai Model; que els requadres 3×3 fan **2,97 × 2,97** Uds., amb una separació entre ells de **0,03** (és a dir, que les dimensions dels requadres, preses intereixos, és exactament de **3,00 × 3,00** Uds.); que, en conseqüència, les caselles fan **0,99 × 0,99** Uds., i no pas les **0,96 × 0,96** Uds. deduïbles de **TG**. Res de més cert, però és que tampoc ens interessava que l'usuari pogués fer clic exactament sobre la línia de separació entre caselles (¿sobre quina de les dues -o de les quatre- havia volgut fer diana?), raó per la qual vam decidir invalidar aquests clics tan fronterers i l'obligar-lo a afinar la punteria. I ara sí, tanquem l'incís esperant que almenys haurà servit perquè el lector no s'estranyi de veure, en el codi de la funció **AREES**, que els 9 elements **G** de la llista **GG** referida a files (F1 i F2), és **(list (mapcar '- (list 0 K) TG) (mapcar '+ (list 8 K) TG))**, amb enters **K** de **0** a **8**.
Definides les àrees **GG**, la funció **INPUNT** serà executada tres vegades (si la segona no produeix error), visualitzant en pantalla els requeriments següents:
- Amb F1:
1 - *Assenyaleu (dreta) la fila que vulgueu moure:*
2 - *Assenyaleu (dreta) on voleu situar-la:*
3 - *Assenyaleu (dreta) on voleu situar la fila desplaçada:*
- Amb F3:
1 - *Assenyaleu (dreta) la tríada de files que vulgueu moure:*
2 - *Assenyaleu (dreta) on voleu situar-la:*
3 - *Assenyaleu (dreta) on voleu situar la tríada de files desplaçada:*
- Amb C1:
1 - *Assenyaleu (dreta) la columna que vulgueu moure:*
2 - *Assenyaleu (dreta) on voleu situar-la:*
3 - *Assenyaleu (dreta) on voleu situar la columna desplaçada:*
- Amb C3:
1 - *Assenyaleu (dreta) la tríada de columnes que vulgueu moure:*
2 - *Assenyaleu (dreta) on voleu situar-la:*
3 - *Assenyaleu (dreta) on voleu situar la tríada de columnes desplaçada:*
Si la segona o tercera entrada ha estat incorrecta (segons la funció **CONTROL** que veurem de seguida) es visualitzarà un avís, seguit de la repetició de la demanda:
  - *No s'hi val> Assenyaleu (dreta) on voleu...*
En definitiva, el que sol·licita **INPUNT** és que s'especifiqui quines files, tríades de files, columnes o tríades de columnes volem permutar, i de quina manera.

Després de cada clic, **ZONA** l'identifica amb la posició que ocupa a la llista **GG** la zona rectangular on s'ha efectuat (procesant F1, C1, F2 o C2, amb valors de 0 a 8; procesant F3 o C3, amb valors de 0 a 2), i **CONTROL** la validarà quan es donin unes condicions simples que, per abreujar encara més, enunciarem referint-nos a zones, sense entrar en si corresponen a files, columnes, tríades de files o de columnes:
- La zona determinada pel clic 1 sempre serà vàlida.
- La zona determinada pel clic 2 no ha de coincidir amb la determinada pel clic 1.
- La zona determinada pel clic 3 no ha de coincidir amb la determinada pel clic 2 (pot coincidir amb la determinada pel clic 1, i llavors permuta el contingut de les zones determinades pels clics 1 i 3, sense afectació de la zona del clic 2).
La quarta condició només s'aplica a F1 i C1:
- Les zones determinades pels clics 2 i 3 han de correspondre a la mateixa tríada que la determinada pel clic 1.
La funció **CONTROL** és més complicada del que caldria esperar, perqué, a més de ser accedida des de F1, F3, C1 i C3, ho és des de F2 i C2 (se n'encarregarà **F//C**, que veurem aviat): en aquestes permutacions, amb només dos clics però també subjectes a les dues primeres condicions, l'última diu el contrari que la quarta de F1 i C1: les zones determinades pels clics 1 i 2 han de correspondre a tríades diferents.

Definides les tres zones, entrarà en escena **PERMUT-G**, que prepara el terreny per a l'acció (de **CAN->VAR** la funció responsable d'aplicar la transformació a ****9**9****), tipificant la permutació a partir de la llista **ZZ** (zones determinades pel primer, segon i tercer clic) creada amb **ZONA**. De fet, aquesta llista de posicions, que en els casos F1 i C1 (o F2 i C2, encara que de moment no ens en ocupem) van de 0 a 8, ens interessa reduir-la a l'escala de la tríada on s'han efectuat els tres clics (0, 1 i 2, com en els casos F3 i C3, en què les tríades tenen aquesta ordenació a escala del SUDOKU-SOLUCIÓ): podríem fer **(setq I (* 3 (/ Z 3)) J (rem Z 3))** i jugar amb **J** en lloc de **Z** (a la pràctica, de fet, no ha calgut recòrrer a **rem**); al final, restituiríem **(setq Z (+ I J))**. Doncs bé: si, després d'haver fet aquesta reducció, la llista de clics és **'(0 2 0)**, per exemple, això únicament pot significar que **el primer clic** s'ha efectuat **a la zona 0** (la posició del contingut que volem moure), **el segon a la zona 2** (la posició on volem situar el contingut mogut, desplaçant el que estava en aquesta posició) i **el tercer a la zona 0** (la posició on volem situar el contigut desplaçat) un altre cop, deixant intacta la zona **1**; i si és **'(0 1 2)**, això només pot significar que **el primer clic** s'ha efectuat **a la zona 0** (la posició del contingut que volem moure), **el segon a la zona 1** (la posició on volem situar el contingut mogut, desplaçant el que estava en aquesta posició) i **el tercer a la zona 2** (la posició on volem situar el contigut desplaçat), i l'únic lloc cap a on pot moure's el contingut d'aquesta última posició serà la zona **0**. El segon exemple és dels que afecta les tres zones, però el primer només n'afecta dues, cosa que es tradueix en l'absència d'una zona (la zona **1**) a la llista o, si ho preferiu, a la presència repetida d'una altra (la zona **0**). És aquesta la raó d'haver afirmat que **'(0 2 0)**, en barrejar una posició de procedència (clic **1**) amb dues de destinació (clics **2** i **3**) sols podia representar clics; si no, la repetició no tindria sentit.

Però la descripció de la situació resultant la podem fer de dues maneres. Potser la més intuitiva serà mitjançant llistes en què l'ordre dels elements correspongui a aquesta situació (la transformada) i el seu valor a l'ordre que ocupaven en la situació original: així, el resultat del primer exemple el representaríem amb la llista **'(2 1 0)** (la zona **0** queda ocupada pel contingut procedent de la zona **2**, la zona **1** queda ocupada pel contingut procedent de la zona **1** -no s'ha vist afectada- i la zona **2** queda ocupada pel contingut procedent de la zona **0**); el resultat del segon, amb la llista **'(2 0 1)** (la zona **0** queda ocupada pel contingut procedent de la zona **2**, la zona **1** queda ocupada pel contingut procedent de la zona **0** i la zona **2** queda ocupada pel contingut procedent de la zona **1**). Tot i ser la que adoptem a **CAN->VAR**, farem esment d'una altra que convindrà utilitzar en passar a **FES-PUBLIC**. Consisteix a usar llistes en què l'ordre dels elements correspongui a la situació original i el seu valor a l'ordre que ocuparan en la situació transformada: així, el resultat del primer exemple el representaríem amb la llista **'(2 1 0)**, com abans (el contingut de la zona **0** passarà a ocupar la zona **2**, el contingut de la zona **1** passarà a ocupar la zona **1** -no s'ha vist pas afectat- i el contingut de la zona **2** passarà a ocupar la zona **0**); però el resultat del segon, amb la llista **'(1 2 0)**, ja no coincideix amb l'altre sistema de representació (el contingut de la zona **0** passarà a ocupar la zona **1**, el contingut de la zona **1** passarà a ocupar la zona **2** i el contingut de la zona **2** passarà a ocupar la zona **0**). Si, atenent el que indiquen els valors de les llistes, convenim en anomenar el primer sistema "PROCEDÈNCIA" (en la llista transformada, els valors indiquen les posicions de procedència) i el segon "DESTINACIÓ" (en la llista original, els valors indiquen les posicions de destinació), la funció **PERMUT-G** definirà la correspondència entre les llistes de clics (**J**) i les de permutació de continguts de zones (**JJ**), en codi de PROCEDÈNCIA (en el quadre hem inclòs el codi de DESTINACIÓ perquè l'usuari copsi coincidències i disparitats amb el de PROCEDÈNCIA, i les tingui en consideració més endavant):

Clics 1, 2 i 3	PROCEDÈNCIA	DESTINACIÓ	Denominació usual
- - - -	**'(0 1 2)**	**'(0 1 2)**	referència 123
'(1 2 1) / **'(2 1 2)**	**'(0 2 1)**	**'(0 2 1)**	permutació 132
'(0 1 2) / **'(1 0 1)**	**'(1 0 2)**	**'(1 0 2)**	permutació 213

'(0 2 1) **'(2 1 0)** **'(2 0 1)**	**'(1 2 0)**	**'(2 0 1)**	permutació 231
'(0 1 2) **'(1 2 0)** **'(2 0 1)**	**'(2 0 1)**	**'(1 2 0)**	permutació 312
'(0 2 0) **'(2 0 2)**	**'(2 1 0)**	**'(2 1 0)**	permutació 321

Pel que fa a **CAN->VAR**, únicament afegirem que: mitjançant les funcions **TR-9**9** (que utilitzarem diverses vegades, en l'àmbit de **CANON->VARIANT**) i **TRANSPOSAR** (que utilitzarem moltes més en tot el programa), efectuarem una transposició de files i columnes que permetrà unificar el tractament de C1 i C3 (també C2) amb el de F1 i F3 (també F2), i mitjançant **F=3** (també usada en tot el programa), reconstruirem ****9**9**** per tríades senceres, en comptes de recórrer a **subst**, substituint només unes files determinades (recurs que ens podem permetre perquè totes les files són diferents), que és el procediment estàndard.

En **CANON->VARIANT**, les respostes **RESP "F2"** i **"C2"** pertoquen a la funció **F//C**, que té una estructura similar a **F/C**, tret que **INPUNT** s'executa dues vegades i no tres, i que la funció responsable d'aplicar la transformació a ****9**9****, en lloc de ser **CAN->VAR** es **PERM-BIN**, molt més senzilla. Però la singularitat de **F//C** en relació a **F/C** i a la resta de funcions específiques de les demés respostes **RESP** (**SIM** per a FS i CS, **TRANSPOSAR** per a TR i **2-9V** per a PV), és que la possibilitat d'utilitzar les permutacions F2 i C2, mantenint ****9**9**** l'estatus de SUDOKU-SOLUCIÓ, no està garantida (la postil·la entre parèntesis a les opcions corresponents del menú ja ens ho advertia: *fins a 216 permutacions, però no sempre és possible*). Més aviat és una excepció a la regla, perquè està condicionada a l'existència de dues files o columnes (de tríades diferents) amb idèntica composició en els primers, segons i tercers tercets (evidentment, els valors comuns han d'estar permutats de forma que no hi hagi coincidències en columnes o files), i d'aquesta inusual circumstància en dóna fe la funció **EXPLORA**. Si **EXPLORA** detecta l'existència d'almenys un parell de files o columnes així, treu un missatge com aquest, que correspon a la SOLUCIÓ CANÒNICA (que, entre altres particularitats que anirem veient, té la propietat que totes les columnes són bescanviables amb dues de diferentes tríades verticals):

Aquesta VARIANT ESCOLLIDA permet permutar les columnes 1ª i 4ª, 1ª i 7ª, 2ª i 5ª, 2ª i 8ª, 3ª i 6ª, 3ª i 9ª, 4ª i 7ª, 5ª i 8ª, i 6ª i 9ª.

Altrament, el missatge hagués estat:

Aquesta VARIANT ESCOLLIDA no permet permutar columnes de tríades diferents.

Per no interferir amb l'exposició ràpida del tractament atorgat als deu operadors, justificarem aquesta condició (necessària i suficient) al final del capítol, just després del codi en negreta. En aquesta mena d'apèndix, que obrirem amb el còmput de SUDOKUs-SOLUCIó efectuat per Felgenhauer i Jarvis, també hi trobareu tot el que fa al nombre de variants que podem arribar a formar, fins i tot amb permutació de requadres 3×3 sencers, transformació reservada a determinats casos, com F2 i C2, i que finalment hem optat per no incloure en el repertori publicitat per **VARIACIONS**.

Les simetries respecte a la fila (FS) o columna (CS) central es resoldran amb **SIM**, que no fa més que invertir les files de ****9**9**** amb **reverse**, prèvia reducció del cas FS al CS (mitjançant **TR-9**9** i **TRANSPOSAR**) i posterior restitució. Quant a TR, l'acció es limita a la transposició directa de ****9**9****, mitjançant **TRANSPOSAR**.

Quan volguem canviar determinats valors de les caselles per uns altres, mitjançant PV, entrarem en un règim iteratiu, en què se'ns anirà sol·licitant que marquem dos valors, fent clic sobre els nombres **1**, **2**, **3**... **9** en un caseller 3×3 que apareixerà centrat a la pantalla, entre les finestres SOLUCIÓ CREADA, COPIADA o CANÒNICA i la VARIANT ESCOLLIDA, acció que provocarà la seva permutació en les 18 caselles de la segona finestra on apareguin. La primera vegada, començarà demanant

CLIC (centre) sobre el nombre a substituir:

i, tot seguit,

CLIC (centre) sobre el nombre que el substituirà:

Però la segona (i posteriors) començarà amb

CLIC (centre) sobre el nombre a substituir <sortir>:

per tal de permetre la sortida de PV, quan els canvis a la finestra dreta ja hagin assolit els objectius previstos. Tot i que sembla que alguna cosa no encaixa prou bé, si parlem d'una correspondència biunívoca (però no necessàriament simètrica) dintre del conjunt dels enters **1**, **2**, **3**... **9**: com definirem aquesta correspondència a base de permutacions amb parells de valors? Suggerim aquest senzill procediment:

- Si la substitució només afecta dos valors, per exemple

  1 2
  ↓ ↓
  2 1, ens limitarem a permutar l'**1** amb el **2**.

- Si la substitució afecta tres valors, per exemple

  1 2 3
  ↓ ↓ ↓
  2 3 1, permutarem l'**1** amb el **2** i després amb el **3**.

- Si la substitució afecta quatre valors, per exemple

  1 2 3 4
  ↓ ↓ ↓ ↓
  2 3 4 1, permutarem l'**1** amb el **2**, després amb el **3** i després amb el **4**.

  .....

- Si la substitució afecta tots nou valors, per exemple

  1 2 3 4 5 6 7 8 9
  ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
  2 3 4 5 6 7 8 9 1, permutarem l'**1** amb el **2**, després amb el **3**, després amb el **4**... després amb el **8** i després amb el **9**.

- Si hi ha cicles diferents, els identificarem i tractarem separadament, com s'ha descrit. Per exemple, si hem de fer

  1 2 3 4 5 6 7 8 9
  ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
  2 3 4 1 6 7 5 9 8

  identificarem els tres cicles

  1 2 3 4 5 6 7 8 9
  ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
  2 3 4 1 6 7 5 9 8 i permutarem l'**1** amb el **2**, després amb el **3** i després amb el **4**; a continuació, permutarem el **5** amb el **6** i després amb el **7**; finalment, permutarem el **8** amb el **9**.

La funció **2-9V** s'ocupa de tot això. Abans d'entrar en el règim iteratiu controlat per **while**, la funció **TECLAT** s'encarregarà de dibuixar el caseller central 3×3 per als valors **1**, **2**, **3**... **9** que haguem de designar, donar-li un fons gris que connecti visualment la finestra SOLUCIÓ CREADA, COPIADA o CANÒNICA i la VARIANT ESCOLLIDA, situar a la casella superior-esquerra de la primera un cursor quadrat que caldrà esborrar (**2-9V** no el necessita, però sí **OMPLE-SUDOKU**, funció que també accedeix a **TECLAT** i on és imprescindible tant per a la SOLUCIÓ CREADA com per a la COPIADA), i posar ordenadament els valors **1**, **2**, **3**... **9** en les caselles, mitjançant **MASCARA** (el codi d'aquesta funció no el presentarem fins el capítol següent, però sí que podem avançar que l'argument és una llista amb els nou valors que han d'aparèixer a les caselles; quant al nom, potser no massa afortunat, es refereix a la missió que té, en el context de l'opció A -SOLUCIÓ CREADA-, de permetre veure únicament els valors compatibles amb les caselles ja emplenades). A partir d'aquí, en cada iteració de **while**, les dues execucions de la funció **INVALOR**, que (anàlogament a **INPUNT** sobre les caselles de la finestra dreta) convida a fer clic sobre alguna de les caselles del caseller central, en combinació amb la funció **TECLA-P** que retorna el seu ordinal **N**, que només inicialment coincideix amb el valor que s'hi exhibeix. Aquests valors es guarden a la llista auxiliar **MASC**, d'on amb cada clic es captura el valor **(nth (1- N) MASC)**, assignat primer a **N1** i després a **N2**, cosa que permet actualitzar la llista, bescanviant-hi aquests elements, i renovar l'aspecte de la màscara fent **(MASCARA MASC)**. Però no només el caseller central 3×3 va acusant les substitucions, sinó la finestra VARIANT ESCOLLIDA, perquè els canvis s'estenen a la pseudomatriu ****9**9**** i es visualitzen fent **(GRAF-9*9)**.

Més d'un lector s'haurà preguntat per què no s'elaborava més la funció **2-9V**, per exemple, permetent que l'usuari manifestés per quins nous valors calia substituir els actuals **1**, **2**, **3**... **9**. Del tot normal, perquè l'autor també s'ho va preguntar,

un cop **2-9V** va quedar resolta en la forma que té ara. Les adaptacions no haurien estat massa complicades, però finalment va decidir deixar-la així, convençut que l'aparent tosquedat donava més flexibilitat d'ús:
- Quan l'usuari no tingui clar quines substitucions l'interessen, li resultarà més senzill anar provant interactivament (els canvis queden immediatament reflectits en la VARIANT ESCOLLIDA) amb permutacions simples (binàries). A cada pas, a més, la correspondència amb la situació inicial es pot copsar observant la posició de cada valor en el caseller central 9×9 (perquè una comparació directa de caselles homòlogues de la SOLUCIÓ CREADA, COPIADA o CANÒNICA i la VARIANT ESCOLLIDA només serà pertinent si abans no hem aplicat cap transformació, fora de la pròpia PV).
- Per contra, quan sàpiga quines substitucions convenen, no suposarà cap molèstia fer el desglossament en cicles (seguint el procediment recomanat línies enrera) sobre el mateix paper on les hagi anotat.

Recuperant la globalitat de **CANON->VARIANT** després d'haver-nos entretingut en les funcions que serveixen l'acció dels operadors, i havent dit ja que les invitacions a seguir-ne aplicant a ****9**9**** se succeiran mentre no donem una resposta nul·la (*sortir*), sols queda precisar que cada iteració acaba amb una crida **(GRAF-9*9)** que actualitza la finestra VARIANT ESCOLLIDA. En acabat, se'ns demana el vist-i-plau a

Us sembla bé com a solució de treball (S/N)? <S>:

i, si no hi ha **CONF**irmació, tornarem a començar des de zero, amb la possibilitat d'optar un altre cop entre una SOLUCIÓ CREADA, COPIADA o la CANÒNICA, d'acord amb l'esquema **(while (/= CONF "Si") ... (PREPTEXT-9*9) ... (CANON->VARIANT))**.

Si ara tornem als prolegòmens de **FES-PUBLIC** i concretament a la pàgina 91, dèiem que, després de la justificació de l'ús de solucions normalitzades, a càrrec de **NORMTEXT-9*9**, i abans de l'exposició de les normes de funcionament amb **FES-PUBLIC**, a càrrec d'**EXPLICACIÓ**, la primera d'aquestes funcions obrava la substitució de la SOLUCIÓ CREADA, COPIADA o CANÒNICA de l'esquerra per una versió normalitzada de la VARIANT ESCOLLIDA de la dreta (canvi reflectit en el peu de la figura, que passava a anomenar-se V. E. NORMALITZADA), exhibint gràficament la modificació de ****9*9****. Doncs bé: abans d'anar al codi de totes les funcions instrumentals, caldrà parlar de **VAR->NORM** i de **PRE-NORM->VAR** (**SEGUIR**, **GRAF-9*9** i **RETOL-1** ja les coneixem), que són les que faltaven (amb les auxiliars **ORDENA-3** i **PERMUT-N**) i s'ocupen d'això.

La funció **VAR->NORM** redefineix ****9*9****, convertint aquesta pseudomatriu 9×9 en la versió normalitzada de la VARIANT ESCOLLIDA ****9***9****, acció amb la qual es perdrà la informació relativa a la SOLUCIÓ CREADA o COPIADA (òbviament, si l'opció era A, la SOLUCIÓ CANÒNICA no es perd). Amb l'ajut d'**ORDENA-3** (que en realitat s'hauria d'anomenar **PREORDENA-3**, perquè l'únic que fa és identificar l'element interMEDi, que després serà situat entre el MÍNim i el MÀXim), reordena les files dels tres requadres 9×3 i finalment reordena els requadres del sudoku. Disortadament, haurem d'estendre'ns més en relació a **PRE-NORM->VAR**, no pas perquè la problemàtica sigui molt complexa sinó perquè, després de diversos canvis d'orientació (des de la idea inicial de disposar de totes les solucions normalitzades, utòpica hipòtesi en què sols faltava expandir-les, a raó de **1.296** variants cada una, per tenir-les totes), és la resolució que ha sortit espessa, amb la primera meitat del codi de **PERMUT-N** que ha quedat obsoleta (l'havíem deixat per si més endavant la necessitàvem, però no ha estat el cas), i amb el mateix nom **PRE-NORM->VAR**, que suggereix l'existència d'una funció **NORM->VAR** que havia existit (com existeix **VAR->NORM**), però que ja no. En aquella versió primigènia, la pantalla de text proposava això

Per triar un SUDOKU-SOLUCIÓ, entre XXX.XXX.XXX.XXX.XXX.XXX solucions possibles, començareu escollint entre XXX.XXX.XXX.XXX.XXX solucions normalitzades (en què les files de cada tríada estan ordenades numèricament, de baix a dalt, i també les tríades estan ordenades, atenent la fila inferior), i permutareu les files de cada tríada i les tríades de la solució, d'acord amb la convenció següent:
- *Entrant el valor 0 es respecta l'ordenació 1-2-3 original;*
- *entrant el valor 1 adoptareu la permutació 1-3-2;*
- *entrant el valor 2 adoptareu la permutació 2-1-3;*
- *entrant el valor 3 adoptareu la permutació 2-3-1;*
- *entrant el valor 4 adoptareu la permutació 3-1-2;*
- *entrant el valor 5 adoptareu la permutació 3-2-1.*

i, per l'espai deixat (les *X* suggerien magnituds que, en cas de no poder precisar, substituiríem per expressions nominals prou suggestives), queda clar que encara no havíem trobat cap referència a la troballa de Felgenhauer i Jarvis, però que fèiem curt (de fet, només sabíem que la segona quantitat havia de ser igual a la primera dividida per $\mathbf{3!^4 = 1.296}$, ignorant que la primera era **6.670.903.752.021.072.936.960** i la segona, doncs, **5.147.302.277.794.037.760**). Va ser després, en renunciar amb

encert a aquesta via condemnada al fracàs i sospitar que cada SUDOKU-SOLUCIÓ ens l'hauríem de suar, que vam sentir la necessitat de forçar-ne al màxim la capacitat d'expansió (anant a $3!^4 \times 3!^4 \times 2 \times 9!$ = **1.218.998.108.160** variants, i encara alguna més en casos singulars) i, en paral·lel, de fer més còmoda la manera d'indicar-li al programa quin tipus de permutació volíem, fent ús exclusivament de la rateta. Després d'això ja no hauríem d'utilitzar-la més mentre no arribéssim a **FES-PUBLIC**, perquè **VAR->NORM** crearia una nova ****9*9**** a partir de ****9**9****, i el que calia era definir la correspondència entre les files d'ambdues pseudomatrius (en definitiva, entre les coordenades **Y** de caselles homòlogues, perquè les **X** no haurien variat), per tal de poder bastir el SUDOKU-PROBLEMA des de qualsevol de les dues finestres: fent CLIC sobre la V. E. NORMALITZADA o sobre la VARIANT ESCOLLIDA, però veient en qualsevol cas com la casella afectada s'activava en les dues (****V*V**** i ****V**V****).

Per això, en lloc d'una funció **NORM->VAR** que obtingués una llista de tres valors **0**, **1** i **2** en codi de PROCEDÈNCIA, a partir del valor convencional **0**... **5** introduït per l'usuari, ens en convenia una que a partir d'una llista en codi de DESTINACIÓ ens donés un valor enter, més apte per a comparacions i càlculs en la determinació d'un altre enter: la coordenada **Y** transformada de la casella on s'hagués fet clic. Ja no importava que el criteri DESTINACIÓ fos menys intuitiu, perquè en la funció **PRE-NORM->VAR** ja no hi intervindria l'usuari: seria un mòdul que calcularia els 4 paràmetres definidors de la correspondència entre la V. E. NORMALITZADA (****9*9****) i la VARIANT ESCOLLIDA (****9**9****), amb l'ajut de **PERMUT-N** (la part operativa de la qual respon a codi DESTINACIÓ però que encara conserva la primera meitat orientada a llistes en l'altre codi). Aquesta paràmetres, que adoptaran valors **0**... **5**, són:
- **N1**, corresponent a les posicions que ocuparien en ****9**9**** les files **0**, **1** i **2**, del primer requadre 9×3 de ****9*9****, si romanguessin en aquest mateix requadre.
- **N2**, corresponent a les posicions que ocuparien en ****9**9**** les files **3**, **4** i **5**, del segon requadre 9×3 de ****9*9****, si romanguessin en aquest mateix requadre.
- **N3**, corresponent a les posicions que ocuparien en ****9**9**** les **files 6**, **7** i **8**, del tercer requadre 9×3 de ****9*9****, si romanguessin en aquest mateix requadre.
- **N4**, corresponent a les posicions que ocuparan en ****9**9**** els **requadres 9×3 0**, **1** i **2** de ****9*9****.

En el capítol 8 (*Detectar solucions compatibles*) abordarem **FES-PUBLIC** (del 3 al 7 ens dedicarem a **OMPLE-SUDOKU**, sobretot per resoldre la problemàtica de l'opció A). Per ara ens limitarem a fer-ne un esbós, per situar la intervenció dels paràmetres calculats per **PRE-NORM->VAR**: després que **INI-COORDS** hagi inicialitzat ****V*V**** i ****V**V****, assignant a cada element les coordenades de la casella que representa, la funció predefinida **grread** ens permetrà anar fent clic a qualsevol de les dues finestres; les coordenades **X**,**Y** de la casella activada seran processades en **CLIC**, funció en què **N->V** o **V->N** (segons que haguem utilitzat la V. E. NORMALITZADA o la VARIANT ESCOLLIDA) localitzarà la coordenada **Y** de la casella homòloga de l'altra finestra, mitjançant les funcions auxiliars **Y1**, **Y2** i **Y3**; finalment, i mitjançant **ELEMENT** (que, a partir de **X** i de les dues **Y**, localitza el valor de les caselles en ****9*9**** i ****9**9****), **COMMUTA** actualitzarà ****V*V**** i ****V**V**** (on substituirà, en l'element activat, les coordenades per l'enter que ocupa la casella) i **V+-** en treu una còpia, canviant-la de capa (de NORM-0 a NORM-1, i de VAR-0 a VAR-1) per passar de gris a blanc. Fet això, veieu **C:SUDOKULUM** i les seves funcions instrumentals:

```
(defun *** () (terpri) (repeat 78 (prompt "*")))

(defun DIB-3*3 (Z)
  (command "SCP" "DE" "1,1"
           "RECTANG" "-1.485,-1.485" "1.485,1.485")
  (if Z (command "ZOOM" "E" "ZOOM" "0.9X"))
  (command "DESCOMP" "LT"
           "MATRIZ" "0,-1.485" "" "R" 3 1 0.99
           "MATRIZ" "-1.485,0" "" "" 1 3 0.99))

(defun TRIA-G ()
  (command "DESHACER" "M" "VENTANAS" "" "ESPACIOM"
           "ZOOM" "C" "7,4.45" 15.8)
  (repeat 3
    (setq G (1+ G))
    (DIB-3*3 ())
    (command "TEXTO" "MC" "-1,-1" "0.5" "0" "1"
             "TEXTO" "MC" "0,-1" "" "" "2"
```

```
              "TEXTO" "MC" "1,-1" "" "" "3"
              "TEXTO" "MC" "-1,0" "" "" "4"
              "TEXTO" "MC" "0,0" "" "" "5"
              "TEXTO" "MC" "1,0" "" "" "6"
              "TEXTO" "MC" "-1,1" "" "" "7"
              "TEXTO" "MC" "0,1" "" "" "8"
              "TEXTO" "MC" "1,1" "" "" "9"
              "CAMBPROP" "OC" "-1,0" "0,1" "1,0" "0,-1" ""
                         "E" "C" "-0.2,-0.2" "0.2,0.2" "" "O" G ""
              "RECTANG" "1.515,-1.485" "13.485,1.485"
              "COLOR" G "SOMBREA" "S" "LT" "" "COLOR" "PORCAPA"
              "TEXTO" "MC" "7.5,1" "" "" "Cal diferenciar prou"
              "TEXTO" "MC" "7.5,0" "" "" "1-3-5-7-9 de 2-4-6-8"
              "TEXTO" "MC" "7.5,-1" "" "" "i llegir bé aquest text."
              "SCP" "DE" "-1,2.45"))
  (while (not (and (= (car (setq GR (grread))) 3)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (> PX -0.485) (< PX 14.485)
                   (setq G (cond ((and (> PY -10.835) (< PY -7.865)) 252)
                                 ((and (> PY  -7.385) (< PY -4.415)) 253)
                                 ((and (> PY  -3.935) (< PY -0.965)) 254))))))
  (command "DESHACER" "R" "TILEMODE" 1))

(defun PREPGRAF-9*9 ()
  (setq G 251               ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        SVT (getvar "SAVETIME")
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
           "SAVETIME" 0
           "SORTENTS" 127
           "FILLMODE" 1
           "OSMODE" 0
           "CAPA" "E" "PAPER" "I" "~PAPER"
;                 "C" "NOR-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "O" 7        ; ACAD 2000
                  "CR" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "CO" 7     ; ACAD 2004
                  "PAPER,NORM-VAR,NORM-1,VAR-1" "D" "NORM-VAR" ""
           "COLOR" "PORCAPA" "TIPOLIN" "D" "CONTINUOUS" ""
           "LWDISPLAY" 0
           "ESTILO" "STANDARD" "TXT" 0 1 0 "N" "N" "N"
           "TILEMODE" 1 "VENTANAS" "N"
           "SCP" "" "PLANTA" "" "SIMBSCP" "DES" "MODOSOMBRA" "2D"
           "PRESENTACION" "N" "SUDOKU"
           "PRESENTACION" "D" "SUDOKU" "SIMBSCP" "DES")
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvingut/Benvinguda al programa SUDÒKULUM!")
  (prompt "  ****************")
  (***) (***)
  (prompt "\n\n\nSUDÒKULUM no ha estat concebut per resoldre sudokus (tot i que ")
  (prompt "pot usar-se amb\naquest propòsit, renunciant al goig de jugar-hi com ")
  (prompt "Déu mana) sinó per fer-ne.\n\nFunciona sobre AutoCAD 2004, i supera ")
  (prompt "altres productes similars en que la seva\nlentitud exasperant obliga")
  (prompt "rà l'usuari a exercitar la virtut de la paciència...\n\nConvé aclarir")
  (prompt " que SUDÒKULUM no té res a veure amb CURRÍCULUM ni amb ESPÈCULUM.\n\n")
  (repeat 68 (prompt " "))
  (prompt "Joan Colom\n\n\nAbans que res, cal fer quatre ajustos: ")
  (prompt "\n\n1) Amplieu al màxim la finestra de text [Per seguir, polseu ")
  (prompt "qualsevol tecla]")
  (grread)
  (graphscr)
  (prompt "\n.\n2) Deixeu tres línies de text inferiors [Per seguir, polseu ")
  (prompt "qualsevol tecla]")
  (grread)
```

```
  (prompt "\n.\n3) Obriu la pàgina \"Visual.\" i en el requadre \"Elementos de ")
  (prompt "presentación\"\n   deixeu activada únicament la 1ª casella ")
  (prompt "[Per seguir, piqueu en \"Aceptar\"]\n")
  (command "OPCIONES")
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: aquella en què el ")
  (prompt "color GRIS es\n   diferencii bé del BLANC i el NEGRE, i doni bon ")
  (prompt "contrast entre text i fons.")
  (TRIA-G)
  (DIB-3*3 T)
  (command "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
           "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
           "CAPA" "B" "NORM-VAR"
;                 "C" G "NORM-0,VAR-0" "D" "PAPER" ""                  ; ACAD 2000
                  "CO" G "NORM-0,VAR-0" "D" "PAPER" ""                 ; ACAD 2004
           "PRESENTACION" "D" "SUDOKU"
           "VENTANAS" "-1.25,-0.5" "-0.25,0.5"
           "VENTANAS" "0.25,-0.5" "1.25,0.5"
           "ZOOM" "E" "ZOOM" "0.9X"
           "ESPACIOM"
;          "CVPORT" 2                                                  ; ACAD 2000
           "CVPORT" 2 "UCSVP" 0                                        ; ACAD 2004
           "VGCAPA" "I" "VAR-1,VAR-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
;          "CVPORT" 3                                                  ; ACAD 2000
           "CVPORT" 3 "UCSVP" 0                                        ; ACAD 2004
           "VGCAPA" "I" "NORM-1,NORM-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
           "SCP" "" "ESPACIOP"
           "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))

(defun PREPTEXT-9*9 ()
  (textscr)
  (prompt "\n\nAquest programa permet crear un SUDOKU-PROBLEMA a partir d'una ")
  (prompt "SUDOKU-SOLUCIÓ,\na base de decidir quines caselles es faran públiques")
  (prompt " (visibles per al jugador).\nPel que fa a la SUDOKU-SOLUCIÓ:")
  (prompt "\n\nA) Podeu improvisar-la, emplenant casella a casella sota control ")
  (prompt "del programa.")
  (prompt "\n\nB) Podeu adoptar-ne una de coneguda, que caldrà transcriure ")
  (prompt "casella a casella.")
  (prompt "\n\nC) Si la solució us és indiferent podeu usar aquesta, que ")
  (prompt "anomenarem CANÒNICA:")
  (prompt "\n\n                                  9 1 2  3 4 5  6 7 8")
  (prompt   "\n                                  6 7 8  9 1 2  3 4 5")
  (prompt   "\n                                  3 4 5  6 7 8  9 1 2")
  (prompt "\n\n                                  8 9 1  2 3 4  5 6 7")
  (prompt   "\n                                  5 6 7  8 9 1  2 3 4")
  (prompt   "\n                                  2 3 4  5 6 7  8 9 1")
  (prompt "\n\n                                  7 8 9  1 2 3  4 5 6")
  (prompt   "\n                                  4 5 6  7 8 9  1 2 3")
  (prompt   "\n                                  1 2 3  4 5 6  7 8 9")
  (initget "A B C")
  (setq ABC (getkword "\n\nTrieu l'opció A, B o C <C>: ") ABC (if ABC ABC "C"))
  (graphscr))

(defun GRAF-9*9 ()
  (setq J -1)
  (foreach 9*9 (list **9*9** **9**9**)
    (if (< J 0)
        (command "CVPORT" 3 "BORRA" "C" "0,0" "8,8" ""
                 "CVPORT" 2 "BORRA" "C" "0,0" "8,8" "" "CAPA" "D" "NORM-1" "")
        (progn
          (setq J -1)
          (command "CVPORT" 3 "CAPA" "D" "VAR-1" "")))
    (while (< (setq J (1+ J)) 9)
           (setq K -1)
           (while (< (setq K (1+ K)) 9)
                  (command "TEXTO" "MC" (list J K) 0.5 0
                                   (itoa (nth J (nth K 9*9))))))))
```

```
(defun SEGUIR (GRAF)
  (if GRAF (prompt "\n.\n."))
  (prompt "\n                  [Per seguir, polseu qualsevol tecla]")
  (grread))

; L'ús de l'ordre CAPAP (no hi era en ACAD 2000 però sí a les versions posteriors)
; portaria conflictes a OMPLE-SUDOKU (en el control de visibilitat de les diverses
; capes des de les dues finestres, establert a PREPGRAF-9*9 mitjançant VGCAPA). És
; per això que prenem nota de la capa prèvia C i la restituïm finalment, amb CAPA.
(defun RETOL-1 (P R / C)
  (setq C (getvar "CLAYER"))
  (if (wcmatch R "V`.*") (command "BORRA" "C" "-0.85,-0.58" "-0.65,-0.52" ""))
  (command "CAPA" "D" "PAPER" "" "TEXTO" "MC" P 0.05 0 R "CAPA" "D" C ""))

(defun RETOL-2 (T1 T2 T3 T4 T5 T6 T7 T8 T9)
  (RETOL-1 '(0 0.4)  T1) (RETOL-1 '(0 0.3)  T2) (RETOL-1 '(0 0.2)  T3)
  (RETOL-1 '(0 0.1)  T4) (RETOL-1 '(0 0)    T5) (RETOL-1 '(0 -0.1) T6)
  (RETOL-1 '(0 -0.2) T7) (RETOL-1 '(0 -0.3) T8) (RETOL-1 '(0 -0.4) T9))

(defun VARIACIONS ()
  (textscr)
  (repeat 16 (terpri))
  (prompt "D'aquesta SUDOKU-SOLUCIÓ se'n poden treure d'altres, per aplicació ")
  (prompt "reiterada\nde les transformacions següents:")
  (prompt "\n\n\nF1: Permutació de Files d'una mateixa tríada\n    (6 ")
  (prompt "permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).")
  (prompt   "\n\nF2: Permutació de Files de tríades diferents")
  (prompt "\n    (fins a 216 permutacions, però no sempre és possible).")
  (prompt   "\n\nF3: Permutació de tríades de Files (6 permutacions).")
  (prompt "\n\n\nFS: Simetria respecte a la Fila central (es pot fer amb tres ")
  (prompt "F1 i una F3).")
  (prompt "\n\n\nC1: Permutació de Columnes d'una mateixa tríada\n    (6 ")
  (prompt "permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).")
  (prompt "  \n\nC2: Permutació de Columnes de tríades diferents")
  (prompt "\n    (fins a 216 permutacions, però no sempre és possible).")
  (prompt   "\n\nC3: Permutació de tríades de Columnes (6 permutacions).")
  (prompt "\n\n\nCS: Simetria respecte a la Columna central (es pot fer amb tres")
  (prompt " C1 i una C3).")
  (prompt "\n\n\nTR: Transposició de files i columnes.")
  (prompt "\n\n\nPV: Permutació entre Valors 1 a 9 estesa a tot el SUDOKU (tot i")
  (prompt " que muntat com\n    una seqüència de permutacions binàries, hi ha 9!")
  (prompt " = 362.880 possibilitats).\n\n")
  (SEGUIR ()))

(defun TRANSPOSAR (A / LLE LL LE E LS LLS)
  (setq LLE A)
  (while LLE
         (setq LS () LL LLE)
         (foreach L LL
                  (setq E (car L) LE (cdr L)
                        LS (append LS (list E))
                        LLE (if (not (equal LLE LL)) LLE)
                        LLE (if LE (append LLE (list LE)))))
         (setq LLS (append LLS (list LS)))))

(defun AREES (/ G)
  (setq K -1)
  (if (wcmatch E "*column*")
      (if (< F 3)
          (repeat 9 (setq K (1+ K)
                          G (list (mapcar '- (list K 0) TG)
                                  (mapcar '+ (list K 8) TG))
                          GG (cons G GG)))
          (repeat 3 (setq K (1+ K)
                          G (list (mapcar '- (list (* 3 K) 0) TG)
                                  (mapcar '+ (list (+ 2 (* 3 K)) 8) TG))
                          GG (cons G GG))))
```

```
      (if (< F 3)
          (repeat 9 (setq K (1+ K)
                          G (list (mapcar '- (list 0 K) TG)
                                  (mapcar '+ (list 8 K) TG))
                          GG (cons G GG)))
          (repeat 3 (setq K (1+ K)
                          G (list (mapcar '- (list 0 (* 3 K)) TG)
                                  (mapcar '+ (list 8 (+ 2 (* 3 K))) TG))
                          GG (cons G GG)))))
  (setq GG (reverse GG)))

(defun PREVIES ()
  (setq E (if (= (substr RESP 1 1) "F") "fil" "column")
        F (atoi (substr RESP 2 1))
        E (if (= F 1) (strcat E "a") (if (= F 2) E (strcat "tríada de " E "es"))))
  (AREES))

(defun ZONA (P)
  (setq Z () K -1)
  (foreach G GG
    (if (and (not Z) (setq K (1+ K))
             (< (caar G) (car P))  (< (cadar G) (cadr P))
             (< (car P) (caadr G)) (< (cadr P) (cadadr G)))
        (setq Z K)))
  Z)

(defun CONTROL (/ OK)
  (if ZZ
      (if (= Z (car ZZ))      ; Clics 2 i 3: mateixa fila/columna (F1/C1 i F2/C2)
          (setq OK ())        ; o tríada (F3/C3) que el clic 1.
          (if (= F 3)
              (setq OK T)
              (progn
                (setq OK (= F 1))             ; Amb F1/C1, els clics 2 i 3 han de
                (if (or (and (= I 0) (< 2 Z)) ; de fer-se a la mateixa tríada del
                        (and (= I 6) (< Z 6)) ; clic 1. Però amb F2/C2, el clic 2
                        (and (= I 3) (or (< Z 3) (< 5 Z)))) ; ha de fer-se en una
                    (setq OK (= F 2))))))                   ; tríada diferent.
      (setq OK T I (if (< F 3) (* 3 (/ Z 3)) 0))) ; Primer clic.
  OK)

(defun CONTROL (/ OK)
  (if ZZ
      (if (= Z (car ZZ))
          (setq OK ())
          (if (= F 3)
              (setq OK T)
              (progn
                (setq OK (= F 2))
                (if (or (= Z I) (= Z (1+ I)) (= Z (+ 2 I)))
                    (setq OK (= F 1))))))
      (setq OK T I (if (< F 3) (* 3 (/ Z 3)) 0)))
  OK)

(defun INPUNT (MSG / GR Z)
  (prompt MSG)
  (while (not (and (= (car (setq GR (grread))) 3)
                   (ZONA (cadr GR))
                   (CONTROL)))
         (if Z (prompt (strcat "\nNo s'hi val>" (substr MSG 2)))))
  (setq ZZ (cons Z ZZ)))

(defun PERMUT-N (A)
  (if (atom A)
      ; La llista (I J K), en codi PROCEDÈNCIA, s'ha d'interpretar com:
      ; - En posició 0 de la permutada hi l'element I de l'original.
      ; - En posició 1 de la permutada hi l'element J de l'original.
      ; - En posició 2 de la permutada hi l'element K de l'original.
```

```
      (cond ((= A 0)  '(0 1 2))
            ((= A 1)  '(0 2 1))
            ((= A 2)  '(1 0 2))
            ((= A 3)  '(1 2 0))
            ((= A 4)  '(2 0 1))
            (   T     '(2 1 0)))
      ; La llista (I J K), en codi DESTINACIÓ, s'ha d'interpretar com:
      ; - L'element 0 de l'original se situa en posició I a la permutada.
      ; - L'element 1 de l'original se situa en posició J a la permutada.
      ; - L'element 2 de l'original se situa en posició K a la permutada.
      (cond ((equal A '(0 1 2)) 0)
            ((equal A '(0 2 1)) 1)
            ((equal A '(1 0 2)) 2)
            ((equal A '(2 0 1)) 3)
            ((equal A '(1 2 0)) 4)
            (   T   5))))

(defun PERMUT-G (/ L) ; s'exclou el cas (= N 0) de PERMUT-N
  (setq L (mapcar '(lambda (Z) (- Z I)) ZZ))
  (cond ((or (equal L '(1 2 1)) (equal L '(2 1 2)))
         '(0 2 1)) ; cas (= A 1) de PERMUT-N
        ((or (equal L '(0 1 0)) (equal L '(1 0 1)))
         '(1 0 2)) ; cas (= A 2) de PERMUT-N
        ((or (equal L '(0 1 2)) (equal L '(1 2 0)) (equal L '(2 0 1)))
         '(1 2 0)) ; cas (= A 3) de PERMUT-N
        ((or (equal L '(0 2 1)) (equal L '(1 0 2)) (equal L '(2 1 0)))
         '(2 0 1)) ; cas (= A 4) de PERMUT-N
        ((or (equal L '(0 2 0)) (equal L '(2 0 2)))
         '(2 1 0)))) ; cas (= A 5) de PERMUT-N

(defun TR-9**9 (F/C)
  (setq **9**9** (if (= (substr RESP 1 1) (if F/C "F" "C"))
                     **9**9**
                     (TRANSPOSAR **9**9**))))

(defun F=3 (9*9)
  (setq 9**9 ())
  (foreach J JJ
    (setq K (1- (* 3 J)))
    (repeat 3 (setq K (1+ K) 9**9 (cons (nth K 9*9) 9**9))))
  (reverse 9**9))

(defun CAN->VAR (/ 9**9)
  (TR-9**9 T)
  (if (= F 1)
      (progn
        (setq 9**9 **9**9**)
        (foreach J '(0 1 2)
          (setq K (+ I J) **9**9** (subst (nth J JJ) (nth K 9**9) **9**9**)))
        (foreach J '(0 1 2)
          (setq K (+ I J) **9**9** (subst (nth K 9**9) J **9**9**))))
      (setq **9**9** (F=3 **9**9**)))
  (TR-9**9 T))

(defun F/C (/ E F GG JJ ZZ)
  (PREVIES)
  (while (not (setq ZZ ()
                    JJ (INPUNT (strcat T1 E T2))
                    JJ (INPUNT T3)
                    JJ (INPUNT (strcat T4 E T5))
                    JJ (PERMUT-G)))
         (prompt "\nNo has definit cap permutació. TORNA-HI!"))
  (CAN->VAR))

(defun EXPLORA (/ EFD FDE HIG IGH LLL)
  (TR-9**9 T)
  (setq LL ())
  (foreach I '(0 1 2 3 4 5)
```

```
    (setq L (nth I **9**9**)
          EFD (list (nth 4 L) (nth 5 L) (nth 3 L))
          FDE (list (nth 5 L) (nth 3 L) (nth 4 L))
          HIG (list (nth 7 L) (nth 8 L) (nth 6 L))
          IGH (list (nth 8 L) (nth 6 L) (nth 7 L))
          LLL (list (append EFD HIG) (append EFD IGH)
                    (append FDE HIG) (append FDE IGH)))
    (foreach J '(3 4 5 6 7 8)
      (if (>= J (* (1+ (/ I 3)) 3))
          (progn
            (setq K (nth J **9**9**))
            (if (member (member (nth 3 K) K) LLL)
                (setq LL (cons (list I J) LL)))))))
  (TR-9**9 T)
  (setq LL (reverse LL)))

(defun PERM-BIN ()
  (TR-9**9 T)
  (setq J (nth (car ZZ) **9**9**) K (nth (cadr ZZ) **9**9**)
        **9**9** (subst () J **9**9**)
        **9**9** (subst J  K **9**9**)
        **9**9** (subst K () **9**9**))
  (TR-9**9 T))

(defun F//C (/ E F GG JJ ZZ)
  (PREVIES)
  (prompt "\n.\n.\nAquesta VARIANT ESCOLLIDA ")
  (if (EXPLORA)
      (progn
        (prompt (strcat "permet permutar les " E "es"))
        (foreach L LL
          (prompt (strcat " " (itoa (1+ (car L))) "ª i " (itoa (1+ (cadr L))) "ª"
                          (if (equal L (last LL))
                                        "."
                                        (if (equal L (cadr (reverse LL)))
                                            ", i" ","))))))
        (while (not (setq ZZ ()
                          ZZ (INPUNT (strcat T1 "primera " E "a: "))
                          ZZ (INPUNT (strcat T1 "segona "  E "a: "))
                          ZZ (if (< (car ZZ) (cadr ZZ)) ZZ (reverse ZZ))
                          JJ (member ZZ LL)))
               (prompt (strcat "\nAquesta permutació (" (itoa (1+ (car ZZ))) "-"
                               (itoa (1+ (cadr ZZ))) ") no és permesa. TORNA-HI!")))
        (PERM-BIN))
      (prompt (strcat "no permet permutar " E "es de tríades diferents.\n"))))

(defun SIM ()
  (TR-9**9 ())
  (foreach F **9**9** (setq **9**9** (subst (reverse F) F **9**9**)))
  (TR-9**9 ()))

(defun TECLAT ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.2,-0.2" "-0.1,-0.1"
           "COPIA" "LT" "" "-1.0419,0.5919" ""
           "EDITPOL" (setq CURSOR (ssget "_L")) "G" 0.006 ""
           "MATRIZ" "-0.1,-0.1" "" "R" 3 3 0.15 0.15
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "COLOR" G
           "SOMBREA" "S" "C" "-0.22,-0.53" "0.22,0.53" ""
           "COLOR" "PORCAPA")
  (MASCARA L1A9)
  (if (and (= ABC "A") (not **9**9**)) (command "ESPACIOM" "CVPORT" 2)))

(defun TECLA-P (VARS / FI)
  (while (not (and (or (= (car (setq GR (grread))) 3)
                       (setq FI (and VARS (> (strlen MSG) 51)
                                     (or (equal GR '(2 13)) (equal GR '(2 32))))))
```

```
                    (or FI (equal (setq GR (cadr GR) GR (list (car GR) (cadr GR)))
                                  '(0 0) 0.2))
                    (or FI (setq N (cond ((equal GR '(-0.15  0.15) 0.05) 1)
                                         ((equal GR '( 0     0.15) 0.05) 2)
                                         ((equal GR '( 0.15  0.15) 0.05) 3)
                                         ((equal GR '(-0.15  0   ) 0.05) 4)
                                         ((equal GR '( 0     0   ) 0.05) 5)
                                         ((equal GR '( 0.15  0   ) 0.05) 6)
                                         ((equal GR '(-0.15 -0.15) 0.05) 7)
                                         ((equal GR '( 0    -0.15) 0.05) 8)
                                         ((equal GR '( 0.15 -0.15) 0.05) 9))))
                    (or VARS (= ABC "B") (member N LL))))))

(defun INVALOR (MSG) (prompt MSG) (setq N ()) (TECLA-P T) N)

(defun 2-9V (/ CURSOR MASC N1 N2 N-F N-9*9)
  (TECLAT)
  (command "BORRA" CURSOR "")
  (setq MASC L1A9)
  (while (INVALOR (strcat "\n.\n." T6 "a substituir" (if N1 " <sortir>" "") ": "))
         (setq N1 (nth (1- N) MASC)
               N2 (nth (1- (INVALOR (strcat T6 "que el substituirà: "))) MASC))
         (if (/= N1 N2)
             (progn
               (command "BORRA" SS "")
               (setq MASC (subst  0 N1 MASC)
                     MASC (subst N1 N2 MASC)
                     MASC (subst N2  0 MASC))
               (MASCARA MASC)
               (setq N-9*9 ())
               (foreach F **9**9**
                 (setq N-F ())
                 (foreach E F
                   (setq N-F (cons (if (= E N1) N2 (if (= E N2) N1 E)) N-F)))
                 (setq N-9*9 (cons (reverse N-F) N-9*9)))
               (setq **9**9** (reverse N-9*9))
               (command "ESPACIOM")
               (GRAF-9*9)
               (command "ESPACIOP"))))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" "" "ESPACIOM"))

(defun CANON->VARIANT (/ T1 T2 T3 T4 T5 T6 RESP OK)
  (graphscr)
  (setq T1 "\n   Assenyaleu (dreta) la "
        T2 " que vulgueu moure: "
        T3 "\n   Assenyaleu (dreta) on voleu situar-la: "
        T4 "\n   Assenyaleu (dreta) on voleu situar la "
        T5 " desplaçada: "
        T6 "\n   CLIC (centre) sobre el nombre " RESP "")
  (while RESP
         (if (/= (substr (getvar "LASTPROMPT") 1 35)
                 "Aquesta VARIANT ESCOLLIDA no permet")
             (prompt "\n."))
         (initget "F1 F2 F3 FS C1 C2 C3 CS TR PV")
         (setq RESP (getkword (strcat "\n.\nTipus de permutació: F1/F2/F3/FS/"
                                      "C1/C2/C3/CS/TR/PV <sortir>: ")))
         (if (not OK) (progn (setq OK T) (command "ESPACIOM")))
         (if (or (= RESP "F1") (= RESP "F3") (= RESP "C1") (= RESP "C3"))
             (F/C)
             (if (or (= RESP "F2") (= RESP "C2"))
                 (F//C)
                 (if (or (= RESP "FS") (= RESP "CS"))
                     (SIM)
                     (if (= RESP "TR")
                         (setq **9**9** (TRANSPOSAR **9**9**))
                         (if (= RESP "PV") (2-9V))))))
         (GRAF-9*9))
  (initget "Si No")
```

```
  (setq CONF
        (getkword "\n.\n.\nUs sembla bé com a solució de treball (S/N)? <S>: ")
        CONF (if CONF CONF "Si")))

(defun ORDENA-3 ()
  (setq A (caar L) B (caadr L) C (caaddr L)
        D (assoc (max A B C) L) A (assoc (min A B C) L)
        L (subst () D (subst () A L))))

(defun VAR->NORM (/ K L 9**9 A B C D)
  (setq K -1)
  (repeat 3
    (setq L ())
    (repeat 3 (setq K (1+ K) L (cons (nth K **9**9**) L)))
    (ORDENA-3)
    (foreach E L (if E (setq L (list A E D))))
    (setq 9**9 (append 9**9 (list L))))
  (setq K -1 L ())
  (foreach E 9**9 (setq L (cons (cons (caar E) E) L)))
  (ORDENA-3)
  (foreach E L (if E (setq **9*9** (append (cdr A) (cdr E) (cdr D))))))

(defun PRE-NORM->VAR (/ L* L** N F F1 F2 F3 F4)
  (foreach L  (reverse **9*9**) (setq L*  (cons (car L) L*)))
  (foreach L (reverse **9**9**) (setq L** (cons (car L) L**)))
  (setq N1 (- 9 (length (member (nth 0 L*) L**))) F1 (/ N1 3) ; (nth 0 L*) és 1
        N2 (- 9 (length (member (nth 3 L*) L**))) F2 (/ N2 3)
        N3 (- 9 (length (member (nth 6 L*) L**))) F3 (/ N3 3)
        N4 (list F1 F2 F3) N4 (PERMUT-N N4)
        F1 (rem N1 3)
        F2 (- 9 (length (member (nth 1 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 2 L*) L**))) F3 (rem F3 3)
        N1 (list F1 F2 F3) N1 (PERMUT-N N1)
        F1 (rem N2 3)
        F2 (- 9 (length (member (nth 4 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 5 L*) L**))) F3 (rem F3 3)
        N2 (list F1 F2 F3) N2 (PERMUT-N N2)
        F1 (rem N3 3)
        F2 (- 9 (length (member (nth 7 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 8 L*) L**))) F3 (rem F3 3)
        N3 (list F1 F2 F3) N3 (PERMUT-N N3)))

(defun NORMTEXT-9*9 ()
  (prompt "\n\n\nSi anomenem SOLUCIÓ NORMALITZADA aquella en què les files de ")
  (prompt "cada tríada\nestan ordenades numèricament d'avall cap amunt, i les ")
  (prompt "tríades de files també\nestan ordenades atenent la fila inferior, ")
  (prompt "a la finestra de l'esquerra veureu\nla V. E. NORMALITZADA, referida ")
  (prompt "a la VARIANT ESCOLLIDA que resta a la dreta\n(per obtenir-la el ")
  (prompt "programa haurà fet fins a tres permutacions F1 i una F3).")
  (prompt "\n\nJugant amb les solucions normalitzades serà força més ràpid el ")
  (prompt "processat per\ndetectar amb quantes caselles plenes la SUDOKU-SOLUCIÓ")
  (prompt " ja queda determinada.")
  (SEGUIR ())
  (VAR->NORM)
  (PRE-NORM->VAR)
  (graphscr)
  (GRAF-9*9)
  (command "ESPACIOP")
  (RETOL-1 '(-0.75 -0.55) "V. E. NORMALITZADA")
  (prompt "\n\nCal tenir l'arxiu 123456789.txt a la ruta de cerca d'AutoCAD. ")
  (prompt "Si no l'hi teniu,\npodeu copiar-lo en el directori d'aquest dibuix ")
  (prompt "i després seguir com si no res.")
  (SEGUIR ()))

(defun EXPLICACIO ()
  (textscr)
  (prompt "\n\n\n\nPer construir el SUDOKU-PROBLEMA a partir d'aquesta ")
  (prompt "SUDOKU-SOLUCIÓ, haureu de\nseguir els passos següents:")
```

```
  (prompt "\n\n1) Com que els valors quedaran velats (en gris), haureu d'anar ")
  (prompt "fent clic a les\n   caselles que vulgueu convertir en públiques ")
  (prompt "(visibles per al jugador). Els\n   clics poden fer-se sobre la ")
  (prompt "V. E. NORMALITZADA o sobre la VARIANT ESCOLLIDA\n   (esquerra o ")
  (prompt "dreta) però per passar de l'una a l'altra caldrà un altre clic.")
  (prompt "\n\n2) Les caselles públiques quedaran destacades (blanc/negre) ")
  (prompt "en ambdues bandes,\n   però, si hi torneu a fer un clic, ")
  (prompt "passaran de nou a ser secretes (en gris).")
  (prompt "\n\n3) Quan el conjunt de caselles públiques assoleixi les ")
  (prompt "condicions necessàries\n   per a una solució única, es produirà ")
  (prompt "una llarga pausa, durant la qual només\n   el text inferior ")
  (prompt "revelarà activitat. Quan aquesta s'acabi, un informe dirà\n")
  (prompt "   quantes SOLUCIONS NORMALITZADES són compatibles amb la ")
  (prompt "V. E. NORMALITZADA.")
  (prompt "\n\n4) Si n'hi ha més d'una, haureu de seguir fent clic ")
  (prompt "fins a tenir-ne només una.")
  (SEGUIR ()))

(defun C:SUDOKULUM (/ ANG TG L1A9 CONF ABC G I J K L LL LLL M N N1 N2 N3 N4 P
                      PX PY X Y OK POST OSN FIL SRT SVT ECO **9*9** **9**9**)
  (setq ANG -0.5   ; Un angle que giri el YIN-YANG a una velocitat proporcionada a
        TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9))                ; la del cursor.
  (PREPGRAF-9*9)
  (while (/= CONF "Si")
         (repeat 40 (terpri))
         (if (= CONF "No")
             (progn
               (if (> (getvar "CVPORT") 1) (command "ESPACIOP"))
               (command "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" "" "ESPACIOM"
                        "CVPORT" 2 "BORRA" "C" "0,0" "8,8" ""
                        "CVPORT" 3 "BORRA" "C" "0,0" "8,8" "")
               (***)
               (prompt "\n*******************************  NOVA SOLUCIÓ")
               (prompt "  *******************************")
               (***)
               (terpri)))
         (PREPTEXT-9*9)
         (if (= ABC "C")
             (setq **9*9** (list '(1 2 3 4 5 6 7 8 9) '(4 5 6 7 8 9 1 2 3)
                                 '(7 8 9 1 2 3 4 5 6)
                                 '(2 3 4 5 6 7 8 9 1) '(5 6 7 8 9 1 2 3 4)
                                 '(8 9 1 2 3 4 5 6 7)
                                 '(3 4 5 6 7 8 9 1 2) '(6 7 8 9 1 2 3 4 5)
                                 '(9 1 2 3 4 5 6 7 8)))
             (OMPLE-SUDOKU))
         (setq **9**9** **9*9**)
         (if (= ABC "C") (progn (command "ESPACIOM") (GRAF-9*9)))
         (command "ESPACIOP")
         (SEGUIR T)
         (RETOL-1 '(-0.75 -0.55) (strcat "SOLUCIÓ " (cond ((= ABC "A") "CREADA")
                                                          ((= ABC "B") "COPIADA")
                                                          (T "CANÒNICA"))))
         (RETOL-1 '(0.75 -0.55) "VARIANT ESCOLLIDA")
         (VARIACIONS)
         (CANON->VARIANT))
  (textscr)
  (NORMTEXT-9*9)
  (EXPLICACIO)
  (FES-PUBLIC)
  (prompt "\n.\n(Als efectes de revocació, l'execució de SUDÒKULUM ")
  (prompt "compta com una sola ordre.)")
  (command "OSMODE" OSN
           "FILLMODE" FIL
           "SORTENTS" SRT
           "SAVETIME" SVT
           "DESHACER" "F")
  (setvar "CMDECHO" ECO)
  (princ))
```

Tot i que aquesta versió de **C:SUDOKULUM** és la primera i, com és lògic, subjecta a modificacions, les funcions auxiliars que la precedeixen (tret de 5 excepcions: **GRAF-9*9**, **F/C**, **F//C**, **2-9V** i **CANON->VARIANT**, que substituirem d'aquí a 10 capítols) ja es representen amb la seva formulació definitiva, per tal de no haver-les de mostrar actualitzades tantes vegades com canvis puntuals hagin de suportar, cosa que en algun moment pot causar la sensació que hi ha complicacions innecessàries. La mateixa funció principal, tot i haver mirat de donar-li versemblança, buscant un punt d'equilibri entre el plantejament primari descrit a l'inici del capítol i el grau d'elaboració de les funciona auxiliars esmentades, incorpora elements que semblen irrellevants i de moment ho són, com per exemple les variables en funcions de constant **ANG** (que té a veure amb **FES-PÚBLIC**) i **SVT** (que posa a 0, desactivant els guardats automàtics del dibuix, perquè no destorbin l'execució del programa). És per això que recomanem al lector no capficar-se massa a trobar justificació a tots els elements en joc, si més no a hores d'ara, abans de la versió definitiva.

De moment, s'han omès les dues principals funcions subsidiàries de **C:SUDOKULUM**: **OMPLE-SUDOKU** (més endavant passa a anomenar-se **FES-SOLUCIÓ**), objecte dels pròxims cinc capítols (*Etapa prèvia: crear una solució*, *I*, *II*, *III*, *IV* i *V*) i **FES-PUBLIC** (passarà a anomenar-se **FES-SUDOKU**), que anirà prenent forma en els tres següents (*Detectar solucions compatibles*, *Condicions mínimes* i *Solucions compatibles ho són totes les que hi són...*). Ho fem així perquè, a diferència de les presentades, que han anat evolucionant d'una manera mecànica (atenent únicament a les exigències de programació) per adaptar-se als canvis de les dues esmentades, aquests canvis han obeït a raons metodològiques, de criteri i d'estratègia, que al cap i a la fi són les que justifiquen la preparació i publicació d'un document com el que teniu a les mans (deixant de banda que una de les debilitats de l'autor sigui aprofitar aquestes ocasions per treure's alguna espina clavada). Com és natural, quan diem que hem exclòs aquestes dues funcions del codi presentat, també ens referim a les seves funcions auxiliars, per bé que caldrà aclarir algunes qüestions al respecte:

- **F=3**, **TECLAT** i **TECLA-P** són accedides des dels àmbits d'**OMPLE-SUDOKU** i **FES-PUBLIC**, però com que també ho són des d'algunes de les altres funcions que, formant part de l'entramat bàsic de **C:SUDOKULUM** no depenen de les anteriors i és per això que figuren entre el codi detallat més amunt, hem decidit incorporar-les per partida doble: al present capítol i als que calgui per la seva adscripció a **OMPLE-SUDOKU** o **FES-PUBLIC**.
- **MASCARA** seria una a incloure en el grup precedent, per inscriure's en l'òrbita d'**OMPLE-SUDOKU** i a la vegada ser accedida des de **2-9V** (mitjançant **TECLA** i també directament), funció no vinculada a l'anterior. Però es dóna la circumstància que concentra una bona part de protagonisme en la deriva metodològica narrada en els cinc capítols següents (**OMPLE-SUDOKU**) i també esdevindrà necessària (tot i que la "mascara" amb les 3×3 cel·les per als valors 1... 9 compatibles romangui invisible) en la funció successora de **FES-PUBLIC** (**FES-SUDOKU**), quan **C:SUDOKULUM** experimenti el desdoblament que es descriu en el capítol *Un nou camí i també una drecera*. Com que en aquest cas seria un disbarat oferir la versió definitiva de **MASCARA** o una gairebé definitiva (com hem fet amb **CANON->VARIANT**), hem optat per prescindir-ne de moment i esperar al pròxim capítol, presentant en societat una versió minimalista a partir de la qual la discusió metodològica l'anirà modelant i donant gruix.
- **COMPLET-9*9**, **ACT-LLL**, **ELEMENT**, **INI-COORDS**, **INI-LLL** i **RETOL-2** no formen part de l'entramat primari de **C:SUDOKULUM** però, accedides tant des de l'àmbit d'**OMPLE-SUDOKU** com des del de **FES-PUBLIC**, tampoc hauria estat cap despropòsit afegir-les a les vistes més amunt. Ara bé: com que aquesta inclusió no ens podria eximir de presentar-les de nou en els capítols monogràficament dedicats a aquests àmbits, que és més recomanable per raons d'integritat (i el criteri establert fins ara), hem preferit no ser reiteratius triplicant les aparicions i limitar-nos al segon context... tret de l'última de les citades, en què a risc d'embolicar encara més la troca hem fet just el contrari: posar-la entre les auxiliars de **C:SUDOKULUM**, tot i no figurar-hi, per proximitat onomàstica i funcional amb **RETOL-1** i per no multiplicar la presència d'un totxo insubstancial amb 9 referències a aquesta. Quan finalment efectuem la recomposició, aplegant els fragments de codi tractats separadament, ja hi haurà ocasió d'aplicar una visió més jerarquitzada i global.

Com que ja n'hi ha prou de practicar aquest esport tan poc pedagògic de referir-se a coses de les quals el lector encara no en pot tenir un coneixement precís, donem per acabat el present capítol. La dotzena de pàgines que segueixen només són una mena d'apèndix sobre el nombre de variants que podem formar amb diferents sudokus.

L'any 2005 B. Felgenhauer i F. Jarvis van avaluar en **6.670.903.752.021.072.936.960** el nombre de SUDOKUs-SOLUCIÓ (en el que resta de capítol, en direm sudokus). Si el còmput és exacte, el conjunt està format per més de **5.472** milions de subconjunts disjunts (és a dir, sense intersecció), cada un dels quals amb **1.218.998.108.160** sudokus caracteritzats pel fet de poder-los obtenir de qualsevol altre sudoku del subconjunt per mitjà de permutacions com les descrites. Aquests subconjunts han de ser disjunts perquè, si un sudoku $\mathbf{A_m}$ del subconjunt **A** fos idèntic a un sudoku $\mathbf{B_n}$ del subconjunt **B**, aplicant qualsevol de les permutacions esmentades n'obtindríem un $\mathbf{A_p = B_q}$ que, per definició, també hauria de pertànyer als subconjunts **A** i **B**. En resum: **A** i **B** no poden tenir cap sudoku en comú, perquè si en tinguessin algun els tindrien tots en comú i parlaríem del mateix subconjunt. Tinguem present que, per obtenir un element del subconjunt partint d'un altre n'hi ha d'haver prou amb una de les $\mathbf{3!^3}$ permutacions F1, una de les **3!** F3, una de les $\mathbf{3!^3}$ C1, una de les **3!** C3, una de les **2** TR i una de les **9!** PV (ja s'entén que hi és inclosa la permutació identitat, és a dir, la no-permutació). L'element resultant no dependrà de l'ordre en què s'hagin efectuat les permutacions (en afirmar això, es dóna per sobreentès que la identitat de files, columnes, tríades de files o de columnes, quan permutem posicions amb F1, C1, F3 o C3, la considerem associada als seus valors numèrics; en canvi, la identitat de caselles, quan en permutem els valors numèrics amb PV, la considerarem associada a les seves posicions), tret que n'hagués intervingut alguna de tipus TR, cas en què per arribar al mateix resultat caldria substituir files [o columnes] per columnes [o files] i requadres 9×3 [o 3×9] per requadres 3×9 [o 9×3] en les permutacions posteriors, que passarien a ser de C1 [o F1] a F1 [o C1] i de C3 [o F3] a F3 [o C3]: si repetim permutacions d'un mateix tipus també sortirà un element del subconjunt, perquè l'aplicació successiva de permutacions d'un mateix tipus equival a aplicar una altra permutació d'aquest tipus. I ara que ja hem vist que el nombre de variants sortia del producte $\mathbf{3!^3 \times 3! \times 3!^3 \times 3! \times 2 \times 9! =}$ $\mathbf{= 1.218.998.108.160}$, el lector potser es preguntarà per què, havent-nos molestat a reproduir aquest cardinal de 13 xifres, ens hem limitat a posar **5.472** milions de subconjunts. La resposta és que el quocient entre **6.670.903.752.021.072.936.960** i **1.218.998.108.160** és **5.472.447.994,25**, i no té massa sentit que un altre cardinal no sigui enter. L'explicació cal trobar-la en el fet que no tots els subconjunts tenen el mateix nombre de sudokus, i no en haver usat suposadament en el producte més factors dels necessaris, adduint que s'haurien pogut obviar les permutacions $\mathbf{3!^3 \times 3!}$ corresponents a C1 i C3 perquè, aplicant TR a un sudoku, aplicant F1 o F3 a aquest transposat i tornant a aplicar TR, s'obtenen els mateixos resultats (aquest és, si més no, el camí que hem seguit a **CAN->VAR** < **F/C**): el nombre operadors que considerem és una cosa, i el nombre de variants a què puguin donar lloc una altra; i, si en lloc de pensar en el nombre de permutacions de columnes [o requadres 3×9] aplicables a un sudoku, pensem en el nombre de permutacions de files [o requadres 9×3] aplicables al seu transposat, en ambdós casos ens en sortiran **216** [**6**], tot i que cal rematar el segon supòsit amb una transposició final per fer coincidir els resultats, però aquesta segona transposició ja la tenim recollida en el factor **2**.

En realitat, no havíem tingut en compte que en alguns subconjunts es pot produir intersecció entre els sudokus resultants de permutar valors numèrics amb PV i els de permutar posicions amb la resta d'operadors (fins i tot entre aquests últims), ni tampoc la repercussió de F2 i C2, raó per la qual les quantitats que hem donat eren provisionals i aproximades: després de l'avaluació feta amb B. Felgenhauer, F. Jarvis va establir amb E. Russell que hi havia **5.472.730.538** subconjunts, enter pel qual **6.670.903.752.021.072.936.960** tampoc no és divisible; ni falta que hi fa, perquè d'una banda hi ha subconjunts que superen en nombre d'elements el supòsit estàndard que hem començat presentant (per efecte de F2 i C2) i de l'altra n'hi ha de menys poblats (per la coincidència de resultats aplicant diferents operadors). Com que no pretenem reformular la demostració de Jarvis y Russell sinó limitar-nos a mostrar exemples que avalin aquestes afirmacions, donarem per fet que el nombre de subconjunts més poblats que la mitjana deu ser inferior al de subconjunts menys poblats, atès que **5.472.730.538** > **5.472.447.994,25.** Però abans hem de parlar dels operadors F2, C2, FS i CS, propòsit que aprofitarem per abandonar les farragoses denominacions "tríada de files" i "tríada de columnes" (refiar-se del context per acotar el significat de "tríada" no sembla molt recomanable) i substituir-les per "requadre 9×3" i "requadre 3×9". D'ara endavant usarem "tríades" discrecionalment, per referirnos a tercets de caselles, en files o columnes, dins d'un requadre 3×3.

FS i CS són combinacions de F1, F3, C1 i C3, perquè el primer (simetria entorn de la fila central) equival a una permutació del primer requadre 9×3 amb el tercer, seguida de permutacions de la primera i la tercera fila en cadascun dels requadres 9×3, i el segon (simetria entorn de la columna central) equival a una permutació del primer requadre 3×9 amb el tercer, seguida de permutacions de la primera i la tercera columna en cadascun dels requadres 3×9, de forma que els sudokus obtinguts ja figuraven en el subconjunt. Però els operadors F2 i C2 (permutació de files o columnes situades en requadres diferents) tenen dues característiques singulars:

1) En general, dues files [columnes] situades en diferents requadres 9×3 [3×9] no són permutables: si les permutem, la matriu 9×9 deixa de ser un sudoku.
2) Fins i tot quan siguin permutables, el sudoku que en resulti serà diferent de tots aquells obtinguts amb la resta d'operadors (**1.218.998.108.160** o menys) i, per tant, s'ha de contabilitzar a part (pel cap baix parlem de doblar el nombre de sudokus del subconjunt, perquè cal considerar la combinació amb tots ells).

Considerem dos requadres 9×3 (la notació és prou explícita i no cal comentar-la):

a_{n3}	b_{n3}	c_{n3}	d_{n3}	e_{n3}	f_{n3}	g_{n3}	h_{n3}	i_{n3}
a_{n2}	b_{n2}	c_{n2}	d_{n2}	e_{n2}	f_{n2}	g_{n2}	h_{n2}	i_{n2}
a_{n1}	b_{n1}	c_{n1}	d_{n1}	e_{n1}	f_{n1}	g_{n1}	h_{n1}	i_{n1}

a_{m3}	b_{m3}	c_{m3}	d_{m3}	e_{m3}	f_{m3}	g_{m3}	h_{m3}	i_{m3}
a_{m2}	b_{m2}	c_{m2}	d_{m2}	e_{m2}	f_{m2}	g_{m2}	h_{m2}	i_{m2}
a_{m1}	b_{m1}	c_{m1}	d_{m1}	e_{m1}	f_{m1}	g_{m1}	h_{m1}	i_{m1}

Si, per exemple, intercanviem les files que comencen amb els elements a_{m1} i a_{n2} ($m,n = 1,2,3$ i $m \neq n$):

- En el primer quadre 3×3 del requadre 9×3 **m**, els tercets a_{m2}, b_{m2}, c_{m2} i a_{m3}, b_{m3}, c_{m3} determinen els elements del tercet a_{m1}, b_{m1}, c_{m1} a substituir, i el mateix passa amb el primer quadre 3×3 del requadre 9×3 **n**, on els tercets a_{n1}, b_{n1}, c_{n1} i a_{n3}, b_{n3}, c_{n3} determinen els elements del tercet a_{n2}, b_{n2}, c_{n2}.
- Així que els elements del tercet a_{m1}, b_{m1}, c_{m1} i els del tercet a_{n2}, b_{n2}, c_{n2} han de ser els mateixos (altrament, ni el primer tercet no encaixaria en el primer quadre del requadre 9×3 **n** ni el segon tercet no encaixaria en el primer quadre del requadre 9×3 **m**, perquè hi hauria elements repetits en ambdós quadres 3×3) però no ordenats igual (dos tercets idèntics comportarien repeticions en les tres primeres columnes), o sigui que els elements han d'estar permutats. Peró, de les 6 permutacions dels tres elements, només valen les 3 obtingudes per rotació (**abc**, **bca**, **cab**), perquè a les seves inversions (**cba**, **acb**, **bac**) hi haurà sempre algun element en la mateixa posició.
- Ara bé: quan es donés aquesta circumstància no hauríem de bescanviar únicament els tercets esmentats, perquè estem parlant d'intercanvi de files senceres; així que l'exigència requerida als dos tercets de referència, a_{m1}, b_{m1}, c_{m1} i a_{n2}, b_{n2}, c_{n2}, també s'ha d'estendre als altres dos parells de tercets. Prenent **m1** com a fila de referència,

a_{m1}	b_{m1}	c_{m1}	d_{m1}	e_{m1}	f_{m1}	g_{m1}	h_{m1}	i_{m1}

  les 8 files **n2**

a_{n2}	b_{n2}	c_{n2}	d_{n2}	e_{n2}	f_{n2}	g_{n2}	h_{n2}	i_{n2}

  intercanviables amb ella (expressades per l'element idèntic en la fila **m1**) són:

b_{m1}	c_{m1}	a_{m1}	e_{m1}	f_{m1}	d_{m1}	h_{m1}	i_{m1}	g_{m1}
b_{m1}	c_{m1}	a_{m1}	e_{m1}	f_{m1}	d_{m1}	i_{m1}	g_{m1}	h_{m1}
b_{m1}	c_{m1}	a_{m1}	f_{m1}	d_{m1}	e_{m1}	h_{m1}	i_{m1}	g_{m1}
b_{m1}	c_{m1}	a_{m1}	f_{m1}	d_{m1}	e_{m1}	i_{m1}	g_{m1}	h_{m1}
c_{m1}	a_{m1}	b_{m1}	e_{m1}	f_{m1}	d_{m1}	h_{m1}	i_{m1}	g_{m1}
c_{m1}	a_{m1}	b_{m1}	e_{m1}	f_{m1}	d_{m1}	i_{m1}	g_{m1}	h_{m1}
c_{m1}	a_{m1}	b_{m1}	f_{m1}	d_{m1}	e_{m1}	h_{m1}	i_{m1}	g_{m1}
c_{m1}	a_{m1}	b_{m1}	f_{m1}	d_{m1}	e_{m1}	i_{m1}	g_{m1}	h_{m1}

(No cal dir que arribaríem a conclusions anàlogues per a l'intercanvi de columnes situades en requadres 3×9 diferents: només caldría que substituíssim la matriu 9×9 que consideràvem per la seva transposada, que és un sudoku del mateix subconjunt.)

Vet aquí perquè hem dit que els sudokus obtinguts de l'intercanvi de dues files o columnes situats en requadres 9×3 o 3×9 diferents no poden coincidir, en general, amb cap dels sudokus definits per les permutacions descrites abans: hem vist que aquesta permutació de files equival a fer permutacions circulars en cada tercet, de sentit contrari en cada fila (quan en un tercet de la fila **m1**, per exemple, el primer element i el segon es desplacen una casella cap a la dreta i el tercer se'n desplaça dues cap a l'esquerra, a la homòloga de la fila **n2** el segon element i el tercer es desplacen una casella cap a l'esquerra i el primer element se'n desplaça dues cap a la dreta), moviment que no té res a veure amb l'intercanvi de columnes senceres dins d'un mateix requadre 3×9 (perquè dels nou elements de cada columna un s'ha desplaçat a l'esquerra, un altre a la dreta i els demés no s'han mogut); en general, tampoc amb l'intercanvi de requadres 3×9 sencers (perquè els elements del tercet no canvien de requadre, pel que fa a les files **m1** i **n2**) ni tampoc amb l'intercanvi de files senceres dins d'un mateix requadre 9×3 (les files **m2** i **m3** no es mouen, i les **n1** i **n3** tampoc) o de requadres 9×3 sencers (**m2** i **m3** romanen en el seu requadre, i **n1** i **n3** en el seu); i, pel que fa a les permutacions de columnes, *mutatis mutandis* podríem aplicar la mateixa argumentació.

Però tota norma general té les seves excepcions. I, si recordeu aquell sudoku que anomenàvem SOLUCIÓ CANÒNICA (perquè no hagueu d'anar passant fulls, el reproduïm a l'esquerra), encara us proposem una variant més singular (centre) amb els elements diferents destacats en negreta, que anomenarem SOLUCIÓ PANÒNICA per no desentonar: gentilici d'una regió de Centreeuropa, amb el qual cert entomòleg va batejar una espècie endèmica de papallona però també la seva filla, que anys a venir seria la gran mecenes dels *boppers* novaiorquesos, us la tornareu a trobar en el capítol 6.

9 1 2	3 4 5	6 7 8	9 **7 8**	3 **1 2**	6 **4 5**	9 **7 8**	3 **1 2**	6 **4 5**
6 7 8	9 1 2	3 4 5	6 **4 5**	9 **7 8**	3 **1 2**	6 **4 5**	9 **7 8**	3 **1 2**
3 4 5	6 7 8	9 1 2	3 **1 2**	6 **4 5**	9 **7 8**	1 2 3	4 5 6	7 8 9
8 9 1	2 3 4	5 6 7	8 9 **7**	2 3 **1**	5 6 **4**	8 9 **7**	2 3 **1**	5 6 **4**
5 6 7	8 9 1	2 3 4	5 6 **4**	8 9 **7**	2 3 **1**	5 6 **4**	8 9 **7**	2 3 **1**
2 3 4	5 6 7	8 9 1	2 3 **1**	5 6 **4**	8 9 **7**	2 3 **1**	5 6 **4**	8 9 **7**
7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	3 **1 2**	6 **4 5**	9 **7 8**

L'anomenada SOLUCIÓ CANÓNICA té una particularitat: els tercets homòlegs de les columnes 1ª, 4ª i 7ª, de les columnes 2ª, 5ª i 8ª, i de les columnes 3ª, 6ª i 9ª són permutacions circulars i, per tot el que hem estat dient, podríem intercanviar les columnes 1ª i 4ª, les columnes 2ª i 5ª o les columnes 3ª i 6ª, però efectuant tots tres intercanvis el resultat serà igual que si haguéssim bescanviat en bloc el requadre 3×9 constituït per les columnes 1ª, 2ª i 3ª pel requadre constituït per les columnes 4ª, 5ª i 6ª (execució de l'opció C3 amb permutació binària del primer i segon requadre 3×9); alternativament, podríem intercanviar les columnes 1ª i 7ª, les columnes 2ª i 8ª o les columnes 3ª i 9ª, però si efectuem tots tres intercanvis el resultat serà igual que si haguéssim bescanviat en bloc el requadre constituït per les columnes 1ª, 2ª i 3ª pel requadre constituït per les columnes 7ª, 8ª i 9ª (execució de l'opció C3 amb permutació binària del primer i tercer requadre 3×9); alternativament, podríem intercanviar les columnes 4ª i 7ª, les columnes 5ª i 8ª o les columnes 6ª i 9ª, però si efectuem tots tres intercanvis el resultat serà el mateix que si haguéssim bescanviat en bloc el requadre 3×9 constituït per les columnes 1ª, 2ª i 3ª pel requadre constituït per les columnes 7ª, 8ª i 9ª (execució de l'opció C3 amb permutació binària del primer i tercer requadre). No cal dir que, executant la segona tríada d'intercanvis després de la primera, l'efecte seria el mateix de l'opció C3 amb una permutació ternària que deixés en primer lloc el tercer requadre 3×9, en segon lloc el primer i en tercer lloc el segon; ni que, executant la primera tríada d'intercanvis després de la segona, l'efecte seria el mateix de l'opció C3 amb una permutació ternària que deixés en primer lloc el segon requadre 3×9, en segon lloc el tercer i en tercer lloc el primer. Tanmateix, res d'això no és extrapolable a les files: la pretensió d'executar l'opció F2 hauria de ser contestada amb un avís de l'estil de *Aquesta SUDOKU-SOLUCIÓ no permet de permutar files de tríades diferents* (utilitzaríem el mot *tríades* que, com és lògic en aquest context, es refereix a *tríades de files*, en comptes del més ambigu *requadre* o del més llarg *requadre 9×3*).

La particularitat del segon sudoku és que els tercets homòlegs de les files 1ª, 4ª i 7ª, de les files 2ª, 5ª i 8ª, i de les files 3ª, 6ª i 9ª també són permutacions circulars i, per tot el que hem estat dient, podríem intercanviar les files 1ª i 4ª, les files 2ª i 5ª o les files 3ª i 6ª, però si efectuem tots tres intercanvis el resultat serà el mateix que si haguéssim intercanviat en bloc el requadre 9×3 constituït per les files 1ª, 2ª i 3ª pel requadre constituït per les files 4ª, 5ª i 6ª (execució de l'opció F3 amb permutació binària del primer i segon requadre 9×3); alternativament, podríem intercanviar les files 1ª i 7ª, les files 2ª i 8ª o les files 3ª i 9ª, però si efectuem tots tres intercanvis el resultat serà el mateix que si haguéssim intercanviat en bloc el requadre 9×3 constituït per les files 1ª, 2ª i 3ª pel requadre constituït per les files 7ª, 8ª i 9ª (execució de l'opció F3 amb permutació binària del primer i tercer requadre); alternativament, podríem intercanviar les files 4ª i 7ª, les files 5ª i 8ª o les files 6ª i 9ª, però si efectuem tots tres intercanvis el resultat serà el mateix que si haguéssim intercanviat en bloc el requadre constituït per les files 1ª, 2ª i 3ª pel requadre 9×3 constituït per les files 7ª, 8ª i 9ª (execució de l'opció F3 amb permutació binària del primer i tercer requadre). No cal dir que, executant la segona tríada d'intercanvis després de la primera, l'efecte seria el mateix de l'opció F3 amb una permutació ternària que deixés en primer lloc el tercer requadre 9×3, en segon lloc el primer i en tercer lloc el segon; ni que, executant la primera tríada d'intercanvis després de la segona, l'efecte seria el mateix de l'opció F3 amb una permutació ternària que deixés en primer lloc el segon requadre 9×3, en segon lloc el tercer i en tercer lloc el primer. Al marge d'això, també té les possibilitats d'intercanvi de columnes de la SOLUCIÓ CANÒNICA, de forma que en la PANÒNICA seria aplicable qualsevol de les opcions F2 i C2, però no les dues (F2 i després C2, o a l'inrevés), tret que l'aplicació reiterada de F2 es traduís en una permutació de requadres 9×3 sencers o que l'aplicació reiterada de C2 es traduís una permutació de requadres 3×9 sencers, casos virtualment equivalents a la seva no utilització: si només es permuten una o dues files per requadre 9×3, varia la composició dels tercets verticals homòlegs d'aquests requadres, que deixarà de ser igual (ho podem veure en la tercera representació de la pàgina precedent, on la permutació de les files 1ª i 7ª de la SOLUCIÓ PANÒNICA anul·la totes les possibilitats de permutar columnes homòlogues), i si només es permuten una o dues columnes per requadre 3×9, varia la composició dels tercets horitzontals homòlegs, que deixarà de ser igual.

Un corol·lari del que hem argumentat és que, si dues files [columnes] de requadres 9×3 [3×9] diferents són intercanviables amb una mateixa fila [columna] del tercer requadre, aquestes primeres files [columnes] també són intercanviables entre si, formant un tercet de files [columnes] permutables en cercle (millor en triangle). Un altre n'és que, si dues de les files [columnes] d'un requadre 9×3 [3×9] són intercanviables per dues files [columnes] d'un altre requadre 9×3 [3×9], les files [columnes] restants d'aquests requadres també són intercanviables. I un altre n'és que, si les tres files [columnes] d'un requadre 9×3 [3×9] són intercanviables per les d'un altre requadre 9×3 [3×9] i les d'aquest són intercanviables per les del tercer requadre 9×3 [3×9], les files [columnes] del primer i del tercer requadres també són intercanviables, formant un tercet de requadres 9×3 [3×9] permutables en cercle o, més ben dit, en triangle. Concatenant els dos últims enunciats, encara obtindríem una conseqüència de més abast: si dues files [columnes] d'un requadre 9×3 [3×9] són intercanviables per dues de les files [columnes] d'un altre requadre 9×3 [3×9] i dues d'aquest segon requadre (no és necessari que siguin les mateixes) ho són per dues de les files [columnes] del tercer requadre 9×3 [3×9], llavors les tres files [columnes] de cadascun dels requadres 9×3 [3×9] són intercanviables per les de qualsevol dels altres dos requadres. (Atès l'advertiment final del paràgraf precedent, ja haureu suposat que el recurs als claudàtors tenia caràcter excloent: files [alternativament, columnes]; 9×3 [alternativament, 3×9]).

Però el nostre propòsit no es pas teoritzar sobre el nombre i varietats de sudokus en què sigui pertinent considerar aquesta casuística (empresa per a la qual no hi ha motivació ni preparació, que tot s'ha de dir) sinó, atès que per pura xamba hem trobat un sudoku paradigmàtic en relació a les permutacions F2 i C2 (no se'n poden donar més, de possibilitats, perque si, per exemple, en postuléssim l'existència d'algun en què la 1ª fila fos permutable amb la 4ª i 5ª, entraríem en contradicció amb l'exigència que les dues últimes tinguéssin igual composició per tercets, en pertànyer al mateix requadre 9×3), calcular quants sudokus podem obtenir en aquest cas concret. Com que en primera posició pot romandre-hi la 1ª fila o haver-hi anat la 4ª o la 7ª, en segona posició pot romandre-hi la 2ª fila o haver-hi anat la 5 o la 8ª, i en tercera posició pot romandre-hi la 3ª fila o haver-hi anat la 6ª o la

9ª, hi ha $\mathbf{3^3}$ combinacions de primers requadres 9×3. Com que en quarta posició pot romandre-hi la 4ª fila o haver-hi anat la 7ª o la 1ª (tret de la que s'hagi quedat en primera posició), en cinquena posició pot romandre-hi la 5ª fila o haver-hi anat la 8ª o la 2ª (tret de la que s'hagi quedat en segona posició), i en sisena posició pot romandre-hi la 6ª fila o haver-hi anat la 9ª o la 3ª (tret de la que s'hagi quedat en tercera posició), hi ha $\mathbf{2^3}$ segons requadres 9×3 possibles. No cal especular sobre els tercers requadres 9×3 possibles, perquè cadascuna de les seves files haurà quedat determinada per les precedents, així que els sudokus que podem obtenir amb permutacions F2 seran $\mathbf{3^3 \times 2^3 = 6^3 = 216}$, dels quals hauríem de deduir $\mathbf{3! = 6}$ (els formats per les files originals 1ª, 2ª, 3ª, 4ª, 5ª, 6ª, 7ª, 8ª i 9ª; 1ª, 2ª, 3ª, 7ª, 8ª, 9ª, 4ª, 5ª i 6ª; 4ª, 5ª, 6ª, 1ª, 2ª, 3ª, 7ª, 8ª i 9ª; 4ª, 5ª, 6ª, 7ª, 8ª, 9ª, 1ª, 2ª i 3ª; 7ª, 8ª, 9ª, 1ª, 2ª, 3ª, 4ª, 5ª i 6ª, i 7ª, 8ª, 9ª, 4ª, 5ª, 6ª, 1ª, 2ª i 3ª) que ja podien obtenir-se permutant els requadres 9×3 del sudoku original mitjançant F3. Però deixant-ho així cometríem una greu omissió, ja que la permutació identitat (la primera de les enumerades: 1ª, 2ª, 3ª, 4ª, 5ª, 6ª, 7ª, 8ª i 9ª) l'hem de mantenir en tots els operadors. Podríem introduir l'esmena escrivint $\mathbf{216 - 5 = 211}$, però preferim expresar-ho en la forma $\mathbf{1 + (216 - 6) = 211}$, pel que veurem tot seguit. Repetint ara el raonament amb columnes i requadres 3×9, acabaríem dient que els sudokus diferents que podem obtenir amb permutacions C2 també són **211**, però a l'hora d'avaluar per quant hem de multiplicar el nombre de combinacions degudes a la resta d'operadors (els **1.218.998.108.160** elements que, ja des de bon principi, atribuíem a la major part de subconjunts de sudokus) cal tenir en compte dues coses:

- Una vegada escollides les permutacions corresponents a F1, F3, C1, C3 i PV, i si transposem o no el sudoku (TR), tenim **210** variants pel que fa a F2, **210** més pel que fa a C2, i l'abstenció en relació a aquests operadors (permutació identitat, única comú a F2 i C2).
- Si, havent aplicat alguna de les $\mathbf{210 + 1 = 211}$ transformacions F2, les opcions a fer-ho amb qualsevol de les $\mathbf{210 + 1 = 211}$ transformacions C2 seguissin obertes, les possibilitats serien $\mathbf{211 \times 211 = 44.521}$, però com que la primera acció barra el pas a l'aplicació de la segona, i viceversa, caldrà escollir una única acció (F2 o C2) entre una oferta de $\mathbf{210 + 210 + 1 = 421}$.

Tanmateix tampoc no és cert que el nombre de variants de la SOLUCIÓ CANÒNICA sigui $\mathbf{1.218.998.108.160 \times 211}$ ni que el de la PANÒNICA sigui $\mathbf{1.218.998.108.160 \times 421}$, i no pas a causa dels últims factors sinó del comú, que en el primer cas serà menor i en el segon encara ho serà més. Pero abans d'entrar en aquesta problemàtica és imperatiu que establim una notació més breu per referir-nos a les tipologíes de permutació entre elements, siguin files en un requadre 9×3, columnes en un de 3×9, o bé requadres 9×3 o 3×9 en un sudoku; altrament, el discurs seria massa farragós. En tot el que queda de capítol, anomenarem 123 el sudoku de referència (o, si ho voleu, la permutació identitat); 132, 213 o 321 les permutacions binàries en què el segon element i el tercer, el primer element i el segon, o el primer element i el tercer, respectivament, intercanvien les posicions; 231 o 312 les permutacions ternàries (obtenibles encadenant dues de les precedents) en què la 1ª posició és ocupada pel segon element, la 2ª pel tercer i la 3ª pel primer, o bé la 1ª posició és ocupada pel tercer element, la 2ª pel primer i la 3ª pel segon. Posem-nos-hi!

Pel que fa a la SOLUCIÓ CANÒNICA, que figura representada en primer lloc (123), les cinc variants que segueixen són precisament aquelles on, en dos o tres dels requadres 3×9, les tres columnes s'han permutat en bloc per les d'un altre, raó per la qual coincideixen amb les permutacions C3 i han estat excloses del còmput, perquè fins aquí el que hem dit de la SOLUCIÓ PANÒNICA és aplicable a la CANÒNICA. Les denominacions corresponen al tipus de permutació C3, i la lletra H es refereix a la direcció d'intercanvi (quan li arribi el torn a la PANÒNICA, en què a banda de l'intercanvi de columnes podrem intercanviar files, també usarem la lletra V). Però no ens hem molestat a representar aquests sudokus com a simple il·lustració del que dèiem més amunt, sinó per poder copsar millor el que direm a propòsit de les permutacions ternàries (231-H i 312-H), tot i que (com passarà massa sovint) el text se'ns quedi en una pàgina i les representacions hagin saltat a la següent. Perquè el cop de tisores inflingit a C2 (les **216** possibilitats s'han quedat en **211** d'originals) no serà l'única retallada procedent: si observeu els requadres 9×3 de les pemutacions esmentades, us adonareu que també s'hauria arribat als mateixos resultats aplicant a tots ells permutacions de files F1 (permutacions 231 als de la variant 231-H i permutacions 312 als de la variant 312-H), fet que autoritza a assestar una altra tisorada, però aquesta vegada a C3 que, de les **6** permutacions possibles (incloent-hi la identitat), passa a tenir-ne només **4** d'originals. Però

tampoc no s'atura aquí aquest frenesí retallador, perquè aquestes dues actuacions F1 també són prescindibles, i passem de **216** permutacions a només **214** d'originals. ¿No endevineu mitjançant quin operador podem emular l'acció d'aquests dos F1 que, al seu torn, emulaven l'acció de dos dels 6 C3 que, al seu torn, emulaven l'acció de 5 dels 216 C2?: un operador que fins ara s'havia mantingut en un discret segon terme i que anirà adquirint cada cop més protagonisme; el permutador de valors PV. Comproveu-ho, si no, substituint en 231-H l'**1** pel **4**, el **2** pel **5**, el **3** pel **6**, el **4** pel **7**, el **5** pel **8**, el **6** pel **9**, el **7** per l'**1**, el **8** pel **2** i el **9** pel **3** (podeu fer-ho permutant 1 amb 4 i després amb 7, 2 amb 5 i després amb 8, i 3 amb 6 i després amb 9), i substituint en 312-H l'**1** pel **7**, el **2** pel **8**, el **3** pel **9**, el **4** per l'**1**, el **5** pel **2**, el **6** pel **3**, el **7** pel **4**, el **8** pel **5** i el **9** pel **6** (podeu fer-ho permutant 1 amb 7 i després amb 4, 2 amb 8 i després amb 5, i 3 amb 9 i després amb 6):

9 1 2	3 4 5	6 7 8	9 1 2	6 7 8	3 4 5	3 4 5	9 1 2	6 7 8
6 7 8	9 1 2	3 4 5	6 7 8	3 4 5	9 1 2	9 1 2	6 7 8	3 4 5
3 4 5	6 7 8	9 1 2	3 4 5	9 1 2	6 7 8	6 7 8	3 4 5	9 1 2
8 9 1	2 3 4	5 6 7	8 9 1	5 6 7	2 3 4	2 3 4	8 9 1	5 6 7
5 6 7	8 9 1	2 3 4	5 6 7	2 3 4	8 9 1	8 9 1	5 6 7	2 3 4
2 3 4	5 6 7	8 9 1	2 3 4	8 9 1	5 6 7	5 6 7	2 3 4	8 9 1
7 8 9	1 2 3	4 5 6	7 8 9	4 5 6	1 2 3	1 2 3	7 8 9	4 5 6
4 5 6	7 8 9	1 2 3	4 5 6	1 2 3	7 8 9	7 8 9	4 5 6	1 2 3
1 2 3	4 5 6	7 8 9	1 2 3	7 8 9	4 5 6	4 5 6	1 2 3	7 8 9
	123			**132-H**			**213-H**	
3 4 5	6 7 8	9 1 2	6 7 8	9 1 2	3 4 5	6 7 8	3 4 5	9 1 2
9 1 2	3 4 5	6 7 8	3 4 5	6 7 8	9 1 2	3 4 5	9 1 2	6 7 8
6 7 8	9 1 2	3 4 5	9 1 2	3 4 5	6 7 8	9 1 2	6 7 8	3 4 5
2 3 4	5 6 7	8 9 1	5 6 7	8 9 1	2 3 4	5 6 7	2 3 4	8 9 1
8 9 1	2 3 4	5 6 7	2 3 4	5 6 7	8 9 1	8 9 1	5 6 7	2 3 4
5 6 7	8 9 1	2 3 4	8 9 1	2 3 4	5 6 7	2 3 4	8 9 1	5 6 7
1 2 3	4 5 6	7 8 9	4 5 6	7 8 9	1 2 3	4 5 6	1 2 3	7 8 9
7 8 9	1 2 3	4 5 6	1 2 3	4 5 6	7 8 9	1 2 3	7 8 9	4 5 6
4 5 6	7 8 9	1 2 3	7 8 9	1 2 3	4 5 6	7 8 9	4 5 6	1 2 3
	231-H			**312-H**			**321-H**	

Pel que fa a la SOLUCIÓ PANÒNICA (123), a la pàgina següent figura seguida per les deu variants on, en dos o tres dels requadres 3×9 o 9×3, les tres columnes o files s'han permutat en bloc per les d'un altre, raó per la qual coincideixen amb les permutacions C3 o F3 i han estat excloses del còmput. Cal aclarir que la variant amb el peu TRS, que figura en setè lloc, no té a veure amb aquesta problemàtica i s'ha situat en aquella posició per aprofitar el buit que quedava si volíem que les permutacions homònimes (tret de H o M) quedessin arrenglerades verticalment, però no trigarem molt a parlar-ne. Al marge d'aquest TRS, les denominacions corresponen al tipus de permutació C3 o F3, i H o V es refereixen a la direcció d'intercanvi. Però tampoc ara no ens hem molestat a representar aquests sudokus per il·lustrar el que ja havíem dit, sinó per poder copsar millor el que direm a propòsit de les permutacions ternàries 231-V i 312-V (perquè a 231-H i 312-H els podem aplicar, fil per randa, tot el que s'ha dit sobre la SOLUCIÓ CANÒNICA), perquè el cop de tisores inflingit a F2 (les **216** possibilitats s'han quedat en **211** d'originals) no serà l'única retallada procedent: si observeu els requadres 3×9 de 231-V i 312-V, us adonareu que també hauríem arribat als mateixos resultats aplicant a tots ells permutacions de columnes C1 (231 i 312), fet que autoritza a assestar una altra tisorada, però aquesta vegada a F3 que, de les **6** permutacions possibles, passa a tenir-ne només **4** d'originals. Tampoc no s'atura aquí el frenesí retallador, perquè aquestes dues actuacions C1 també són prescindibles, i passem de **216** permutacions a només **214** d'originals, perquè amb PV podem obtenir els mateixos resultats: només cal substituir en 231-V l'**1** pel **2**, el **2** pel **3**, el **3** per l'**1**, el **4** pel **5**, el **5** pel **6**, el **6** pel **4**, el **7** pel **8**, el **8** pel **9** i el **9** pel **7** (podeu fer-ho permutant 1 amb 2 i després amb 3, 4 amb 5 i després amb 6, i 7 amb 8 i després amb 9), i substituir en 312-H l'**1** pel **3**, el **2** per l'**1**, el **3** pel **2**, el **4** pel **6**, el **5** pel **4**, el **6** pel **5**, el **7** pel **9**, el **8** pel **7** i el **9** pel **8** (podeu fer-ho permutant 1 amb 3 i després amb 2, 4 amb 6 i després amb 5, i 7 amb 9 i després amb 8):

9 **7 8**	3 **1 2**	6 **4 5**	9 7 8	6 4 5	3 1 2	3 1 2	9 7 8	6 4 5
6 **4 5**	9 **7 8**	3 **1 2**	6 4 5	3 1 2	9 7 8	9 7 8	6 4 5	3 1 2
3 **1 2**	6 **4 5**	9 **7 8**	3 1 2	9 7 8	6 4 5	6 4 5	3 1 2	9 7 8
8 9 **7**	2 3 **1**	5 6 **4**	8 9 7	5 6 4	2 3 1	2 3 1	8 9 7	5 6 4
5 6 **4**	8 9 **7**	2 3 **1**	5 6 4	2 3 1	8 9 7	8 9 7	5 6 4	2 3 1
2 3 **1**	5 6 **4**	8 9 **7**	2 3 1	8 9 7	5 6 4	5 6 4	2 3 1	8 9 7
7 8 9	1 2 3	4 5 6	7 8 9	4 5 6	1 2 3	1 2 3	7 8 9	4 5 6
4 5 6	7 8 9	1 2 3	4 5 6	1 2 3	7 8 9	7 8 9	4 5 6	1 2 3
1 2 3	4 5 6	7 8 9	1 2 3	7 8 9	4 5 6	4 5 6	1 2 3	7 8 9
123			**132-H**			**213-H**		
3 1 2	6 4 5	9 7 8	6 4 5	9 7 8	3 1 2	6 4 5	3 1 2	9 7 8
9 7 8	3 1 2	6 4 5	3 1 2	6 4 5	9 7 8	3 1 2	9 7 8	6 4 5
6 4 5	9 7 8	3 1 2	9 7 8	3 1 2	6 4 5	9 7 8	6 4 5	3 1 2
2 3 1	5 6 4	8 9 7	5 6 4	8 9 7	2 3 1	5 6 4	2 3 1	8 9 7
8 9 7	2 3 1	5 6 4	2 3 1	5 6 4	8 9 7	2 3 1	8 9 7	5 6 4
5 6 4	8 9 7	2 3 1	8 9 7	2 3 1	5 6 4	8 9 7	5 6 4	2 3 1
1 2 3	4 5 6	7 8 9	4 5 6	7 8 9	1 2 3	4 5 6	1 2 3	7 8 9
7 8 9	1 2 3	4 5 6	1 2 3	4 5 6	7 8 9	1 2 3	7 8 9	4 5 6
4 5 6	7 8 9	1 2 3	7 8 9	1 2 3	4 5 6	7 8 9	4 5 6	1 2 3
231-H			**312-H**			**321-H**		
9 3 6	7 1 4	8 2 5	8 9 7	2 3 1	5 6 4	9 7 8	3 1 2	6 4 5
8 2 5	9 3 6	7 1 4	5 6 4	8 9 7	2 3 1	6 4 5	9 7 8	3 1 2
7 1 4	8 2 5	9 3 6	2 3 1	5 6 4	8 9 7	3 1 2	6 4 5	9 7 8
6 9 3	4 7 1	5 8 2	9 7 8	3 1 2	6 4 5	7 8 9	1 2 3	4 5 6
5 8 2	6 9 3	4 7 1	6 4 5	9 7 8	3 1 2	4 5 6	7 8 9	1 2 3
4 7 1	5 8 2	6 9 3	3 1 2	6 4 5	9 7 8	1 2 3	4 5 6	7 8 9
3 6 9	1 4 7	2 5 8	7 8 9	1 2 3	4 5 6	8 9 7	2 3 1	5 6 4
2 5 8	3 6 9	1 4 7	4 5 6	7 8 9	1 2 3	5 6 4	8 9 7	2 3 1
1 4 7	2 5 8	3 6 9	1 2 3	4 5 6	7 8 9	2 3 1	5 6 4	8 9 7
TRS			**132-V**			**213-V**		
7 8 9	1 2 3	4 5 6	8 9 7	2 3 1	5 6 4	7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3	5 6 4	8 9 7	2 3 1	4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9	2 3 1	5 6 4	8 9 7	1 2 3	4 5 6	7 8 9
9 7 8	3 1 2	6 4 5	7 8 9	1 2 3	4 5 6	8 9 7	2 3 1	5 6 4
6 4 5	9 7 8	3 1 2	4 5 6	7 8 9	1 2 3	5 6 4	8 9 7	2 3 1
3 1 2	6 4 5	9 7 8	1 2 3	4 5 6	7 8 9	2 3 1	5 6 4	8 9 7
8 9 7	2 3 1	5 6 4	9 7 8	3 1 2	6 4 5	9 7 8	3 1 2	6 4 5
5 6 4	8 9 7	2 3 1	6 4 5	9 7 8	3 1 2	6 4 5	9 7 8	3 1 2
2 3 1	5 6 4	8 9 7	3 1 2	6 4 5	9 7 8	3 1 2	6 4 5	9 7 8
231-V			**312-V**			**321-V**		

Però, ¿què és TRS i què hi pinta el sudoku de la dreta? Encara que no ho sembli, tots dos tenen la propietat de transformar-se amb el seu transposat amb un PV que sols afecti 6 valors: a TRS cal permutar el **2** amb el **4**, el **3** amb el **7** i el **6** amb el **8**, i en surt la SOLUCIÓ PANÒNICA (aplicant el mateix PV a la SOLUCIÓ PANÒNICA, surt TRS); en el sudoku de la dreta cal permutar el **4** amb el **5**, el **6** amb el **7** i el **8** amb el **9**), i en sortirà el transposat. Com que aquests PVs emulen l'acció de l'operador TR, el cardinal del subconjunt SOLUCIÓ PANÒNICA es queda en la meitat, però també el de la dreta, que era dels vulgars.

6 4 2	9 1 5	7 8 3
3 7 1	8 6 4	5 2 9
5 8 9	7 3 2	1 4 6
9 1 7	6 8 3	2 5 4
8 6 5	4 2 9	3 7 1
2 3 4	1 5 7	6 9 8
7 9 3	5 4 6	8 1 2
4 2 8	3 7 1	9 6 5
1 5 6	2 9 8	4 3 7

- **Subconjunt estàndard:**

F1		F3		C1		C3		F2 o C2		3×3		TR		PV
216	×	6	×	216	×	6	×	1	×	1	×	2	×	362.880 =

= **1.218.998.108.160 sudokus**

- **Subconjunt SOLUCIÓ CANÒNICA:**

F1		F3		C1		C3		F2 o C2		3×3		TR		PV
(216-2)	×	6	×	216	×	(6-2)	×	(1+216-6)	×	61	×	2	×	362.880 =
= 214	×	6	×	216	×	4	×	211	×	61	×	2	×	362.880 =

= **5.181.483.140.628.480 sudokus**

- **Subconjunt SOLUCIÓ PANÒNICA:**

F1		F3		C1		C3		F2 o C2		3×3		TR		PV
(216-2)	×	(6-2)	×	(216-2)	×	(6-2)	×	[1 + 2(216-6)]	×	295	×	(2-1)	×	362.880 =
= 214	×	4	×	214	×	4	×	421	×	295	×	1	×	362.880 =

= **33.022.859.292.057.600 sudokus**

- **Subconjunt reduït:**

F1		F3		C1		C3		F2 o C2		3×3		TR		PV
216	×	6	×	216	×	6	×	1	×	1	×	1	×	362.880 =

= **609.499.054.080 sudokus**

Tot plegat està molt bé, però ¿què hi pinta aquest nou camp anomenat "3×3", i per què el cardinal dels subconjunts a què pertanyen la SOLUCIÓ CANÒNICA i la PANÒNICA resulten majorats per efecte del seu valor (del qual, a sobre, no s'ens facilita abans la descomposició operativa habitual, que ens podria posar sobre la pista)?

Efectivament, no resulta massa edificant que haguem intentat passar de matuta una transformació inèdita (que, com indica el nom, és una permutació de requadres 3×3 sencers). Però, com que l'autor havia desestimat la seva inclusió en **VARIACIONS** i **CANON->VARIANT**, i d'altra banda la determinació del nombre de variants no era tan immediata com en altres permutacions (a diferència de F2 i C2, que quan concorren sobre un mateix cas -SOLUCIÓ PANÒNICA- són mútuament excloents, la permutació de requadres 3×3 que comparteixen columnes condiciona però no impedeix una posterior permutació de requadres 3×3 que comparteixen files), vam optar per no complicar el discurs desenvolupat fins aquí i posposar la presentació en societat d'aquest nou personatge al moment en què no hi hagués més remei, per la necessitat d'aplegar en el quadre-resum la contribució de tots ells. Igual que les transformacions F2 i C2 no són invariants dels sudokus (en el doble sentit de no ser aplicables a tots els subconjunts en què les altres ho són ni de ser extensives a totes les combinacions possibles de files o columnes de diferents requadres 9×3 o 3×9, fins i tot en els subconjunts on són aplicables) i, malgrat això, les hem tingut en compte a l'hora de calcular el cardinal de cada subconjunt, donarem acollida a les permutacions de requadres 3×3, amb restriccions anàlogues. I, per començar, en delimitarem l'àmbit d'actuació, aclarint i justificant que aquests requadres només són permutables si pertanyen al mateix requadre 9×3 o 3×9, tret del cas trivial que siguin idèntics.

Si considerem que un requadre 3×3 està format per tres tercets horitzontals, per poder-lo permutar amb un altre no situat en el mateix requadre 9×3, els tercets horitzontals homòlegs hauran d'estar compostos pels mateixos 3 valors (l'ordenació pot ser diferent): altrament, els 6 valors que completen les files on se situen els tercets tampoc serien els mateixos, i això impediria que els tercets fossin intercanviables (no serien compatibles amb els 6 valors de les files receptores). I si considerem que el requadre 3×3 està format per tres tercets verticals, per poder-lo permutar amb un altre no situat en el mateix requadre 3×9, els tercets verticals homòlegs hauran d'estar compostos pels mateixos 3 valors (l'ordenació pot ser diferent): altrament, els 6 valors que completen les columnes on se situen els tercets tampoc serien els mateixos, i això impediria que els tercets fossin bescanviables (no serien compatibles amb els 6 valors de les columnes receptores). Si dos requadres 3×3 no comparteixen requadre 9×3 ni requadre 3×9 i són idèntics, són permutables; recíprocament, si són permutables, el valor de cada casella (que és l'únic comú als seus tercets horitzontal i vertical) ha de ser igual al de la seva homòloga (que és l'únic comú als tercets homòlegs horitzontal i vertical, de composició igual als originals respectius), i per tant són idèntics.

Centrant-nos, doncs, en les úniques permutacions possibles entre requadres 3×3:
- Si considerem que els situats en un mateix requadre 9×3 estan formats per tres tercets verticals, els tercets homòlegs han de tenir igual composició (per tal de ser compatibles amb els 6 valors restants de les columnes homòlogues, que també hauran de tenir la mateixa composició), però amb els valors ordenats de forma diferent (òbviament, perquè no es repeteixin valors en les files).
- Si considerem que els situats en un mateix requadre 3×9 estan formats per tres tercets horitzontals, els tercets homòlegs han de tenir igual composició (per tal de ser compatibles amb els 6 valors restants de les files homòlogues, que també hauran de tenir la mateixa composició), però amb els valors ordenats de forma diferent (òbviament, perquè no es repeteixin valors en les columnes).

Des d'aquests requisits, i de manera similar al que passava amb les permutacions F2 i C2, la SOLUCIÓ CANÒNICA i, especialment, la PANÒNICA, en són els paradigmes. Com abans, partirem del supòsit que totes les permutacions detectables sobre la primera representació (permutació identitat 123) constitueixen variants inèdites, i ens limitarem a mostrar ordenadament (permutacions 132, 213, 231, 312 i 321 de requadres 3×3 que comparteixen files -H- o columnes -V-) aquells sudokus que no contribueixen a enriquir el catàleg, perquè ja es poden obtenir amb permutacions F1, C1, F3 o C3.

Pel que fa a la SOLUCIÓ CANÒNICA (123), podeu veure que en tots tres requadres 9×3 són aplicables les 5 permutacions entre requadres 3×3 (els seus tercets verticals homòlegs tenen la mateixa composició), raó por la qual el factor que correspon a aquesta transformació seria, en principi, $\mathbf{6^3 = 216}$. Però, si mireu 231-H i 312-H (representacions on hem volgut aplegar les $\mathbf{6^3 - 4^3 = 152}$ combinacions on hi intervé alguna d'aquestes dues permutacions ternàries, i per això hem interposat entre els requadres 9×3 línies discontínues), us adonareu que els requadres 9×3 adopten una configuració que també hauríem pogut assolir amb F1. Ara bé: de les altres $\mathbf{4^3 = 64}$ combinacions (una de les quals és 123), cal excloure aquelles en què s'ha aplicat la mateixa permutació binària a tots tres requadres 9×3 (132-H, 213-H i 321-H, que ja no són representacions genèriques sinó de sudokus concrets), perquè adopten una configuració que també hauríem pogut assolir amb C3. Així que el factor es quedarà en $\mathbf{64 - 3 = 61}$. De cara a la SOLUCIÓ PANÒNICA, que abordarem tot seguit, serà útil considerar aquest valor desglossat de la manera següent: la SOLUCIÓ CANÒNICA sense afectacions (**1**); les variants que afecten un sol requadre 9×3 ($\mathbf{3 \times 3 = 9}$); les que afecten dos requadres 9×3 ($\mathbf{3 \times 3^2 = 27}$) i les que n'afecten tres ($\mathbf{1 \times 3^3 = 27}$, d'on hem de treure els sudokus 132-H, 213-H i 321-H). Així, doncs: $\mathbf{1 + 9 + 27 + 24 = 61}$.

9 1 2	3 4 5	6 7 8
6 7 8	9 1 2	3 4 5
3 4 5	6 7 8	9 1 2
8 9 1	2 3 4	5 6 7
5 6 7	8 9 1	2 3 4
2 3 4	5 6 7	8 9 1
7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9

123

9 1 2	6 7 8	3 4 5
6 7 8	3 4 5	9 1 2
3 4 5	9 1 2	6 7 8
8 9 1	5 6 7	2 3 4
5 6 7	2 3 4	8 9 1
2 3 4	8 9 1	5 6 7
7 8 9	4 5 6	1 2 3
4 5 6	1 2 3	7 8 9
1 2 3	7 8 9	4 5 6

132-H

3 4 5	9 1 2	6 7 8
9 1 2	6 7 8	3 4 5
6 7 8	3 4 5	9 1 2
2 3 4	5 6 7	8 9 1
8 9 1	2 3 4	5 6 7
5 6 7	8 9 1	2 3 4
1 2 3	4 5 6	7 8 9
7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3

213-H

3 4 5	6 7 8	9 1 2
9 1 2	3 4 5	6 7 8
6 7 8	9 1 2	3 4 5
2 3 4	5 6 7	8 9 1
8 9 1	2 3 4	5 6 7
5 6 7	8 9 1	2 3 4
1 2 3	4 5 6	7 8 9
7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3

231-H

6 7 8	9 1 2	3 4 5
3 4 5	6 7 8	9 1 2
9 1 2	3 4 5	6 7 8
5 6 7	8 9 1	2 3 4
2 3 4	5 6 7	8 9 1
8 9 1	2 3 4	5 6 7
4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9
7 8 9	1 2 3	4 5 6

312-H

6 7 8	3 4 5	9 1 2
3 4 5	9 1 2	6 7 8
9 1 2	6 7 8	2 4 5
5 6 7	2 3 4	8 9 1
2 3 4	8 9 1	5 6 7
8 9 1	5 6 7	2 3 4
4 5 6	1 2 3	7 8 9
1 2 3	7 8 9	4 5 6
7 8 9	4 5 6	1 2 3

321-H

Pel que fa a la SOLUCIÓ PANÒNICA (123), podeu veure que en tots els requadres 9×3 són aplicables les 5 permutacions entre requadres 3×3 (els seus tercets verticals homòlegs tenen la mateixa composició), igual que la SOLUCIÓ CANÒNICA, però també en tots els requadres 3×9 (els seus tercets horitzontals homòlegs tenen la mateixa composició). Malgrat això, el factor que correspon a aquesta transformació no serà $\mathbf{61^2 = 3.721}$ (cosa que passaria si ambdues formes de permutar requadres 3×3 fossin independents, com ho són F1 i C1, o F3 i C3) ni tampoc $\mathbf{1 + (2 \times 60) = 121}$ (cosa que passaria si fossin mútuament excloents, com ho són F2 i C2), sinó que es quedarà en una quantia intermèdia: les **295** possibilitats que figuraven en el quadre-resum. Abans de justificar aquesta xifra, però, convé que descrivim les representacions. Les sis primeres són semblants a les precedents: la SOLUCIÓ PANÒNICA (123) seguida dels sudokus que cal excloure del còmput, en donar uns resultats assolibles amb F1 (231-H i 312-H, representacions genèriques on apleguem les 152 combinacions en què intervé alguna d'aquestes permutacions ternàries, i per això interposem entre els requadres 9×3 línies discontínues) o amb C3 (132-H, 213-H i 321-H, sudokus en què s'ha aplicat la mateixa permutació binària a tots tres requadres 9×3). Tot seguit, les altres sis: repetim la SOLUCIÓ PANÒNICA (123), per tenir-la més a mà (evitant així un buit), i segueixen els sudokus que cal excloure del còmput, en donar uns resultats assolibles amb C1 (231-V i 312-V, representacions genèriques on apleguem les 152 combinacions on hi intervé alguna d'aquestes permutacions ternàries, i per això interposem entre els requadres 3×9 línies discontínues) o amb F3 (132-V, 213-V i 321-V, sudokus en què s'ha aplicat la mateixa permutació binària a tots tres requadres 3×9). Ara ja estem en disposició d'aclarir d'on surt la xifra **295**, però abans oferirem les 12 representacions (i disculpes perquè 3 hagin saltat pàgina):

9 **7 8** 3 **1 2** 6 **4 5** 6 **4 5** 9 **7 8** 3 **1 2** 3 **1 2** 6 **4 5** 9 **7 8** 8 9 **7** 2 3 **1** 5 6 **4** 5 6 **4** 8 9 **7** 2 3 **1** 2 3 **1** 5 6 **4** 8 9 **7** 7 8 9 1 2 3 4 5 6 4 5 6 7 8 9 1 2 3 1 2 3 4 5 6 7 8 9	9 7 8 6 4 5 3 1 2 6 4 5 3 1 2 9 7 8 3 1 2 9 7 8 6 4 5 8 9 7 5 6 4 2 3 1 5 6 4 2 3 1 8 9 7 2 3 1 8 9 7 5 6 4 7 8 9 4 5 6 1 2 3 4 5 6 1 2 3 7 8 9 1 2 3 7 8 9 4 5 6	3 1 2 9 7 8 6 4 5 9 7 8 6 4 5 3 1 2 6 4 5 3 1 2 9 7 8 2 3 1 8 9 7 5 6 4 8 9 7 5 6 4 2 3 1 5 6 4 2 3 1 8 9 7 1 2 3 7 8 9 4 5 6 7 8 9 4 5 6 1 2 3 4 5 6 1 2 3 7 8 9
123	**132-H**	**213-H**
3 1 2 6 4 5 9 7 8 9 7 8 3 1 2 6 4 5 6 4 5 9 7 8 3 1 2 --- 2 3 1 5 6 4 8 9 7 8 9 7 2 3 1 5 6 4 5 6 4 8 9 7 2 3 1 --- 1 2 3 4 5 6 7 8 9 7 8 9 1 2 3 4 5 6 4 5 6 7 8 9 1 2 3	6 4 5 9 7 8 3 1 2 3 1 2 6 4 5 9 7 8 9 7 8 3 1 2 6 4 5 --- 5 6 4 8 9 7 2 3 1 2 3 1 5 6 4 8 9 7 8 9 7 2 3 1 5 6 4 --- 4 5 6 7 8 9 1 2 3 1 2 3 4 5 6 7 8 9 7 8 9 1 2 3 4 5 6	6 4 5 3 1 2 9 7 8 3 1 2 9 7 8 6 4 5 9 7 8 6 4 5 3 1 2 --- 2 3 1 8 9 7 5 6 4 8 9 7 5 6 4 2 3 1 5 6 4 2 3 1 8 9 7 --- 1 2 3 7 8 9 4 5 6 7 8 9 4 5 6 1 2 3 4 5 6 1 2 3 7 8 9
231-H	**312-H**	**321-H**
9 **7 8** 3 **1 2** 6 **4 5** 6 **4 5** 9 **7 8** 3 **1 2** 3 **1 2** 6 **4 5** 9 **7 8** 8 9 **7** 2 3 **1** 5 6 **4** 5 6 **4** 8 9 **7** 2 3 **1** 2 3 **1** 5 6 **4** 8 9 **7** 7 8 9 1 2 3 4 5 6 4 5 6 7 8 9 1 2 3 1 2 3 4 5 6 7 8 9	8 9 7 2 3 1 5 6 4 5 6 4 8 9 7 2 3 1 2 3 1 5 6 4 8 9 7 9 7 8 3 1 2 6 4 5 6 4 5 9 7 8 3 1 2 3 1 2 6 4 5 9 7 8 7 8 9 1 2 3 4 5 6 4 5 6 7 8 9 1 2 3 1 2 3 4 5 6 7 8 9	9 7 8 3 1 2 6 4 5 6 4 5 9 7 8 3 1 2 3 1 2 6 4 5 9 7 8 7 8 9 1 2 3 4 5 6 4 5 6 7 8 9 1 2 3 1 2 3 4 5 6 7 8 9 8 9 7 2 3 1 5 6 4 5 6 4 8 9 7 2 3 1 2 3 1 5 6 4 8 9 7
123	**132-V**	**213-V**

```
7 8 9 | 1 2 3 | 4 5 6
4 5 6 | 7 8 9 | 1 2 3
1 2 3 | 4 5 6 | 7 8 9
      |       |
9 7 8 | 3 1 2 | 6 4 5
6 4 5 | 9 7 8 | 3 1 2
3 1 2 | 6 4 5 | 9 7 8
      |       |
8 9 7 | 2 3 1 | 5 6 4
5 6 4 | 8 9 7 | 2 3 1
2 3 1 | 5 6 4 | 8 9 7
```

231-V

```
8 9 7 | 2 3 1 | 5 6 4
5 6 4 | 8 9 7 | 2 3 1
2 3 1 | 5 6 4 | 8 9 7
      |       |
7 8 9 | 1 2 3 | 4 5 6
4 5 6 | 7 8 9 | 1 2 3
1 2 3 | 4 5 6 | 7 8 9
      |       |
9 7 8 | 3 1 2 | 6 4 5
6 4 5 | 9 7 8 | 3 1 2
3 1 2 | 6 4 5 | 9 7 8
```

312-V

```
7 8 9   1 2 3   4 5 6
4 5 6   7 8 9   1 2 3
1 2 3   4 5 6   7 8 9

8 9 7   2 3 1   5 6 4
5 6 4   8 9 7   2 3 1
2 3 1   5 6 4   8 9 7

9 7 8   3 1 2   6 4 5
6 4 5   9 7 8   3 1 2
3 1 2   6 4 5   9 7 8
```

321-V

Si considerem únicament les permutacions de requadres 3×3 que comparteixen files (permutacions binàries, perquè les ternàries han quedat excloses), el resultat és idèntic a la SOLUCIÓ CANÒNICA: **61** possibilitats desglossades en la pròpia solució sense afectacions (**1**), les variants que afecten un sol requadre 9×3 (**9**), les que afecten dos requadres 9×3 (**27**) i les que n'afecten tres (**24**). Però si considerem, per a cada variant, les possibilitats que resten d'efectuar permutacions entre els requadres 3×3 que comparteixen columnes, que només es donaran entre aquells que no hagin estat afectats per les precedents, el repertori s'amplia: mantenim la pròpia SOLUCIÓ CANÒNICA, sense afectacions (**1**); pel que fa a les **9** variants que afecten un únic requadre 9×3, totes **9** deixen un espai de tres requadres 3×3 en columna (**4** pemutacions, comptant la identitat) i dos espais de dos requadres 3×3 en columna (**2** pemutacions, comptant la identitat), completant **9×4×2×2 = 144** combinacions; pel que fa a les **27** variants que afecten dos requadres 9×3, n'hi ha **9** que deixen un espai de tres requadres (**4** pemutacions) i **18** que en deixen dos de dos requadres (**2** pemutacions), completant **(9×4)+(18×2×2) = 108** combinacions, i pel que fa a les **24** variants que afecten tres requadres 9×3, n'hi ha **18** que deixen un espai de dos requadres (**2** pemutacions) i **6** que no en deixen cap, completant **(18×2)+6 = 42** combinacions. En total, doncs: **1 + 144 + 108 + 42 = 295**.

Per acabar, un comentari a propòsit de l'especulació, iniciada en la pàgina 109, en què xifràvem el cardinal de la major part de subconjunts en **1.218.998.108.160** sudokus i, sabent que el nombre total de sudokus era **6.670.903.752.021.072.936.960** i que el de subconjunts era **5.472.730.538** > **5.472.447.994,25**, valor aquest últim que resultava de dividir la segona quantitat per la primera, pronosticàvem que el nombre de subconjunts més poblats que la mitjana havia de ser inferior al de menys poblats. Tot i que la intenció s'endevinava, calia haver parlat amb més propietat, referint-nos d'una banda al nombre de sudokus (i no al de subconjunts de sudokus), i presentar abans aquesta mitjana (en no fer-ho algú podia creure que parlàvem de **1.218.998.108.160**, valor que no era mitjana de res sinó la moda, estadísticament parlant). Ho farem ara: el cardinal (o nombre de sudokus) mitjà dels subconjunts és **1.218.935.174.261,1** (**6.670.903.752.021.072.936.960** / **5.472.730.538**). I ara, sí: com que és lleugerament inferior al nombre de subconjunts que suposem més abundant (**1.218.998.108.160** és el cardinal d'aquests subconjunts, que anomenem estandard), direm que hi haurà menys sudokus per damunt de la mitjana que per sota; amb tota seguretat, els primers correspondran íntegrament als subconjunts amb més sudokus que els estandard, i els segons correspondran als subconjunts amb menys sudokus que els estandard, perqué en el quadre-resum que obre la pàgina 116 (resum dels casos tractats, que no implica necessàriament que sigui exhaustiu) s'aprecia que les oscil·lacions són més grans que la petita diferència entre **1.218.998.108.160** i **1.218.935.174.261,1**. Això ja s'intuïa a partir de les dades que figuraven en el quadre, perquè el subconjunt SOLUCIÓ PANÒNICA té unes 27.000 vegades més sudokus que els estàndard, però sembla ser únic (l'autor ja ha advertit que aquest no és el seu cavall de batalla); el subconjunt SOLUCIÓ CANÒNICA té unes 4.250 vegades més sudokus que els estàndard, però sembla que n'hi ha d'haver pocs (o també n'ni ha un de sol?); per contra, de subconjunts com l'anomenat reduït, que només té la meitat de sudokus que els estàndard (en poder prescindir de la transformació TR, el resultat de la qual és a l'abast d'un senzill PV) es veu de seguida que n'hi ha d'haver una munió. Probabilísticament, només cal fixar-se en la composició dels tercets de files i columnes dels requadres 3×3, i detectar si tots tres valors o només dos es repeteixen al llarg dels requadres 9×3 o 3×9: en el modestos sudokus dels subconjunts reduïts únicament dos dels tres valors es repeteixen al llarg de cada requadre 9×3 (però aquests dos valors no són els mateixos en els altres dos

requadres 9×3), i també al llarg de cada requadre 3×9 (però aquests dos valors no són els mateixos en els altres dos requadres 3×9); en el(s) subconjunt(s) de la SOLUCIÓ CANÒNICA els tres valors es repeteixen al llarg de cada requadre 9×3 (però aquests valors no són els mateixos en els altres dos requadres 9×3), i al llarg de tots tres requadres 3×9, que tenen aquests valors en comú; en el subconjunt de la SOLUCIÓ PANÒNICA, finalment, els tres valors es repeteixen al llarg de tots tres requadres 9×3, que tenen aquests valors en comú, i al llarg de tots tres requadres 3×9, que també tenen aquests valors en comú.

Abans de cloure aquest capítol volem justificar la manca de correspondència formal entre la versió d'**EXPLORA** que abans hem presentat, entre les altres funcions, i la llista de deu pàgines enrera (8 combinacions de tercets permutats de la fila **m1**,

a_{m1} b_{m1} c_{m1} d_{m1} e_{m1} f_{m1} g_{m1} h_{m1} i_{m1}

que, coincidint amb alguna fila **n2**, n'autoritzaven el bescanvi), que repetim aquí:

b_{m1}	c_{m1}	a_{m1}	e_{m1}	f_{m1}	d_{m1}	h_{m1}	i_{m1}	g_{m1}
b_{m1}	c_{m1}	a_{m1}	e_{m1}	f_{m1}	d_{m1}	i_{m1}	g_{m1}	h_{m1}
b_{m1}	c_{m1}	a_{m1}	f_{m1}	d_{m1}	e_{m1}	h_{m1}	i_{m1}	g_{m1}
b_{m1}	c_{m1}	a_{m1}	f_{m1}	d_{m1}	e_{m1}	i_{m1}	g_{m1}	h_{m1}
c_{m1}	a_{m1}	b_{m1}	e_{m1}	f_{m1}	d_{m1}	h_{m1}	i_{m1}	g_{m1}
c_{m1}	a_{m1}	b_{m1}	e_{m1}	f_{m1}	d_{m1}	i_{m1}	g_{m1}	h_{m1}
c_{m1}	a_{m1}	b_{m1}	f_{m1}	d_{m1}	e_{m1}	h_{m1}	i_{m1}	g_{m1}
c_{m1}	a_{m1}	b_{m1}	f_{m1}	d_{m1}	e_{m1}	i_{m1}	g_{m1}	h_{m1}

Efectivament, la transcripció literal de la llista hauria dut a la versió següent:

```
(defun EXPLORA (/ BCA CAB EFD FDE HIG IGH LLL)
  (TR-9**9 T)
  (setq LL ())
  (foreach I '(0 1 2 3 4 5)
    (setq L (nth I **9**9**)
          BCA (list (nth 1 L) (nth 2 L) (nth 0 L))
          CAB (list (nth 2 L) (nth 0 L) (nth 1 L))
          EFD (list (nth 4 L) (nth 5 L) (nth 3 L))
          FDE (list (nth 5 L) (nth 3 L) (nth 4 L))
          HIG (list (nth 7 L) (nth 8 L) (nth 6 L))
          IGH (list (nth 8 L) (nth 6 L) (nth 7 L))
          LLL (list (append BCA EFD HIG) (append BCA EFD IGH)
                    (append BCA FDE HIG) (append BCA FDE_IGH)
                    (append CAB EFD HIG) (append CAB EFD IGH)
                    (append CAB FDE HIG) (append CAB FDE IGH)))
    (foreach J '(3 4 5 6 7 8)
      (if (>= J (* (1+ (/ I 3)) 3))
          (if (member (nth J **9**9**) LLL) (setq LL (cons (list I J) LL))))))
  (TR-9**9 T)
  (setq LL (reverse LL)))
```

Si allà apareixia sensiblement escurçada (i és aquella la versió que mantindrem), era per una raó ben senzilla: determinada la composició del segon tercet (f_{m1}, d_{m1} i e_{m1}) i del tercer (h_{m1}, i_{m1} i g_{m1}), la del primer (a_{m1}, b_{m1} i c_{m1}) també ho està; d'altra banda, les permutacions del primer diferents de b_{m1}- c_{m1}- a_{m1} i de c_{m1}- a_{m1}- b_{m1} no poden aparèixer en el mateix sudoku, perquè això implicaria coicidències en columna amb a_{m1}- b_{m1}- c_{m1}. Així que vam prescindir de **BCA** i **CAB**.

Etapa prèvia: crear una solució, I (embolica, que fa fort!)

L'autor creu que ha de començar referint-se a la gènesi d'aquest capítol, que de ser-ne un més, pel que fa a extensió, ha anat creixent fins al punt de tornar-se gairebé obligat el desdoblament en cinc subcapítols presidits pel mateix epígraf (*Etapa prèvia: crear una solució*) però diferenciats amb un ordinal (*I*, *II*, *III*, *IV* i *V*) i una il·lustració jocosa (*embolica que fa fort*; *la difícil simplicitat*; *roda el món i torna al Born*; *... o a Camprodon*; *la Seca, la Meca i la vall d'Andorra*) a mode de distintiu. La causa ha estat el canvi gradual d'actitud respecte a l'opció A de **ABC** ("SOLUCIÓ CREADA"): quan va començar a treballar en el tema (*Inventariar les solucions: un cul-de-sac*) ni se la plantejava; després (*Tornem a començar: de la solució al problema*) va sorgir com una exigència de cortesia obligada però que ni de bon tros no pretenia ser el nucli dur d'un programa l'especificitat del qual residia més en la consecució del PROBLEMA a partir de la SOLUCIÓ que en la manera d'obtenir aquesta última; però al final, potser adonant-se de l'escassa solvència d'allò que havia sobrevalorat com a més específic, va acabar trobant deficiències de mètode i d'eficiència en aquesta opció i s'hi ha dedicat fins a deixar-la a un nivell si més no competitiu respecte a ofertes similars.

Abans d'abordar l'estratègia de resolució serà obligat referir-se a l'estructura de dades que reflecteix l'estat d'emplenament de l'escaquer 9×9. Farem servir dues pseudomatrius 9×9 (amb aquest mot seguirem designant les llistes constituïdes per llistes-fila de valors assimilats a elements d'una matriu, que en el cas present estaran formades per 9 llistes de 9 valors 1...9), en què els elements corresponen a les caselles de l'escaquer identificades per l'ordinal de l'element en la fila i el d'aquesta en la pseudomatriu, localitzats d'esquerra a dreta i de baix a dalt:

- ****9*9****, amb només la qual podríem passar, on les caselles buides es representen per les coordenades (els dos ordinals esmentats, guardats en una llista) i les plenes pel valor 1... 9 assignat. La utilitzen totes les opcions.

- **LLL**, que no és indispensable (inicialitzada a **INI-LLL** i actualitzada a **ACT-LLL**), però que convé tenir a mà per estalviar temps, ens indica en cada casella buida (els elements corresponents a les ocupades es representen amb **nil**) quins valors 1... 9 són assignables en primera instància (per no aparèixer en la fila, en la columna ni en el requadre 3×3 on se situa): ho fa en una llista de 9 valors, on surten ordenadament els valors candidats i **nil** en el lloc dels que no ho són. Només la utilitza l'opció A.

I abans de centrar-nos en la problemàtica d'A, dedicarem unes línies a les opcions B i C perquè no sigui dit, tot i ser obvi el seu funcionament a la vista del codi.

Amb C, la SOLUCIÓ CANÒNICA (que ja ve en forma de pseudomatriu 9×9) s'assigna en bloc a ****9*9**** i ****9**9**** (futura VARIANT ESCOLLIDA), immediatament visibles a les respectives finestres (****9*9**** a l'esquerra i ****9**9**** a la dreta) mitjançant la funció **GRAF-9*9**. Pel que fa a B, la conveniència de substituir el teclat per la més còmoda rateta en la introducció dels valors que volem copiar (fent CLIC sobre un teclat numèric virtual que es desplega entre les dues finestres, amb els valors 1... 9 ordenats en un caseller 3×3) i de disposar d'una mira quadrada que emmarqui (en la finestra esquerra) la casella objecte de l'assignació immediata, ha fet que compartís **OMPLE-SUDOKU** amb l'opció A i ambdues participessin d'aquests recursos: la funció **TECLAT** i la variable **CURSOR**. Les diferències respecte aquesta opció són que a B no li cal la funció **TECLA-M** per marcar la casella que volem emplenar, sinó que la mira **CURSOR**, situada inicialment a la casella superior esquerra 1,9, anirà avançant cap a la dreta, casella a casella i saltant de fila en fila cap avall, cada vegada que seleccionem un valor del caseller central 3×3 mitjançant la funció **TECLA-P**, fins a completar l'escaquer 9×9 amb l'emplenament de la casella 9,1 (tot i que en realitat hauríem de parlar d'escaquers perquè les caselles emplenades es visualitzen simultàniament en les dues finestres), i que no hi ha cap control en l'assignació (se suposa que l'usuari ha escollit una SUDOKU-SOLUCIÓ correcta i que la copia bé). El negatiu d'aquestes característiques ens dibuixarà el perfil de l'opció A: discrecionalitat a l'hora d'emplenar una o altra casella, i control de l'assignació de forma que no s'infringeixin les regles el joc. Però abans d'entrar a fons en l'opció, veiem el codi d'**OMPLE-SUDOKU** i de les funcions del seu àmbit:

```
(defun MASCARA (L)
  (setq LL () SS (ssadd))
  (foreach N L1A9
    (if (member N L)
        (progn
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         ((= N 2)     '(0 0.15))
                         ((= N 3)  '(0.15 0.15))
                         ((= N 4) '(-0.15 0))
                         ((= N 5)     '(0 0))
                         ((= N 6)  '(0.15 0))
                         ((= N 7) '(-0.15 -0.15))
                         ((= N 8)     '(0 -0.15))
                         (   T     '(0.15 -0.15)))
                   (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL))))))

(defun TECLAT ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.2,-0.2" "-0.1,-0.1"
           "COPIA" "LT" "" "-1.0419,0.5919" ""
           "EDITPOL" (setq CURSOR (ssget "_L")) "G" 0.006 ""
           "MATRIZ" "-0.1,-0.1" "" "R" 3 3 0.15 0.15
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "COLOR" G
           "SOMBREA" "S" "C" "-0.22,-0.53" "0.22,0.53" ""
           "COLOR" "PORCAPA")
  (MASCARA L1A9)
  (if (and (= ABC "A") (not **9**9**)) (command "ESPACIOM" "CVPORT" 2)))

(defun INI-COORDS ()
  (setq Y -1 PY ())
  (reverse (repeat 9
                 (setq Y (1+ Y) X -1 PX ()
                       PY (cons (reverse (repeat 9
                                                 (setq X (1+ X)
                                                       PX (cons (list X Y) PX))))
                                PY)))))

(defun INI-LLL ()
  (setq LL () LLL ())
  (repeat 9 (setq LL (cons L1A9 LL)))
  (repeat 9 (setq LLL (cons LL LLL))))

(defun COMPLET-9*9 (/ COMP)
  (setq COMP T)
  (foreach L **9*9**
    (if COMP (foreach E L
               (if (and COMP (listp E)) (setq COMP ()))))
  COMP)

(defun ELEMENT (LL) (nth X (nth Y LL)))

(defun TECLA-M ()
  (while (not (and (= (car (setq GR (grread))) 3)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (equal (list PX PY) '(4 4) 4.48)
                   (or (< (- PX (setq X (fix PX))) 0.48)
                       (< (- (setq X (fix (1+ PX))) PX) 0.48))
                   (or (< (- PY (setq Y (fix PY))) 0.48)
                       (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                   (ELEMENT LLL))))
  (setq P (list X Y)))

(defun M->P () (list (+ (* 0.1105 X) -1.1919) (+ (* 0.1105 Y) -0.4419)))
```

```
(defun TECLA-P (VARS / FI)
  (while (not (and (or (= (car (setq GR (grread))) 3)
                       (setq FI (and VARS (> (strlen MSG) 51)
                                     (or (equal GR '(2 13)) (equal GR '(2 32))))))
                   (or FI (equal (setq GR (cadr GR)) GR (list (car GR) (cadr GR)))
                                 '(0 0) 0.2))
                   (or FI (setq N (cond ((equal GR '(-0.15  0.15) 0.05) 1)
                                        ((equal GR '( 0     0.15) 0.05) 2)
                                        ((equal GR '( 0.15  0.15) 0.05) 3)
                                        ((equal GR '(-0.15  0   ) 0.05) 4)
                                        ((equal GR '( 0     0   ) 0.05) 5)
                                        ((equal GR '( 0.15  0   ) 0.05) 6)
                                        ((equal GR '(-0.15 -0.15) 0.05) 7)
                                        ((equal GR '( 0    -0.15) 0.05) 8)
                                        ((equal GR '( 0.15 -0.15) 0.05) 9))))
                   (or VARS (= ABC "B") (member N LL))))))

(defun ACT-LLL (/ X/3 Y/3 LL L J I)
  (setq X/3 (/ X 3) Y/3 (/ Y 3) J -1)
  (foreach F LLL
    (setq L () I -1 J (1+ J))
    (foreach E F
      (setq I (1+ I)
            L (cons (if (= J Y)
                        (if (= I X) () (subst () N E))
                        (if (= I X)
                            (subst () N E)
                            (if (and (= (/ I 3) X/3) (= (/ J 3) Y/3))
                                (subst () N E) E)))
                    L)))
    (setq LL (cons (reverse L) LL)))
  (reverse LL))

(defun ACTUALITZA (/ L LL)
  (setq LL () J -1)
  (foreach E **9*9**
    (setq J (1+ J)
          L (if (= J Y) (subst N P E) E) LL (cons L LL)))
  (setq **9*9** (reverse LL) LLL (ACT-LLL)))

(defun OMPLE-SUDOKU (/ CURSOR P-CURS SS)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq **9*9** (INI-COORDS)
              P-CURS '(-1.1919 0.4419))
       (INI-LLL)
        (while (not (COMPLET-9*9))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP"
                        "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (TECLA-P ())
               (command "ESPACIOM"
                        "CVPORT" 2
                        "CAPA" "D" "NORM-1" ""
                        "TEXTO" "MC" P 0.5 0 (itoa N)
                        "CVPORT" 3
                        "CAPA" "D" "VAR-1" ""
                        "TEXTO" "MC" P 0.5 0 (itoa N)
                        "CVPORT" 2)
               (ACTUALITZA))
        (command "ESPACIOP")
        (setq POST ())))
```

```
      (progn
        (prompt "\n.\n.\nCLIC sobre el nombre que va a la casella marcada: ")
        (setq K 9)
        (repeat 9
          (setq L () J -1 K (1- K))
          (repeat 9
            (setq J (1+ J))
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.110465,0" ""
                     "ESPACIOM"
                     "CVPORT" 2
                     "CAPA" "D" "NORM-1" ""
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "CVPORT" 3
                     "CAPA" "D" "VAR-1" ""
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "ESPACIOP")
            (setq L (append L (list N))))
          (command "DESPLAZA" CURSOR "" "-0.9942,-0.1105" "")
          (setq **9*9** (cons L **9*9**)))))
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))
```

Centrant-nos ja en l'opció A, que ocupa la primera part d'**OMPLE-SUDOKU** (la segona meitat de la pàgina precedent), un cop **TECLAT** hagi completat la presentació SUDOKU creada per **PREPGRAF-9*9**, amb el caseller 3×3 ubicat entre les dues finestres i la mira **CURSOR** en posició de sortida, coincidint visualment amb la casella 1,9 de la finestra esquerra (centrada en el punt de coordenades 0,8 de l'Espai Model, perquè cal recordar que finestres, caseller i mira pertanyen a l'Espai Paper), i després d'inicialitzar les pseudomatrius ****9*9**** (**INI-COORDS**) i **LLL** (**INI-LLL**), arrenca el procés iteratiu que haurà de culminar amb l'obtenció d'una SUDOKU-SOLUCIÓ. A cada iteració tindran lloc tres accions principals:

- Activació d'una casella lliure a la finestra de l'esquerra, a càrrec de l'usuari (**TECLA-M** espera un CLIC en l'Espai **M**odel i n'obté el centre **P** de la casella).
- Aparició en el caseller central dels valors 1... 9 assignables a la casella, a càrrec de SUDÒKULUM (**MASCARA**).
- Selecció del valor d'assignació, a càrrec de l'usuari (**TECLA-P** espera un CLIC en l'Espai **P**aper i n'obté el valor **N**).

Al final de cara iteració **ACTUALITZA** actualitza ****9*9**** i **LLL**, i el procés segueix fins que **COMPLET-9*9** detecta que l'escaquer 9×9 ja està complet.

Si seguiu amb atenció les funcions depenents d'**OMPLE-SUDOKU** no tindreu dificultats a interpretar tots els passos, tret potser de **M->P**, la interposició de la qual en l'expressió **(command ... "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))** convé aclarir. La primera assignació **(setq ... P-CURS '(-1.1919 0.4419))** pot posar-nos sobre la pista del que passa: a diferència de l'opció B, en què la mira quadrada **CURSOR**, que ressalta la casella que anem a emplenar, es desplaça regularment per intèrvals de **0.110465** (mitjana a l'Espai Paper) en ambdues direccions coordenades, a l'opció A resulta més pràctic referir les caselles per les coordenades del seu centre en l'Espai Model, que aproximadament són valors enters (**TECLA-M**), però quan hem de traslladar **CURSOR** d'una posició a la següent cal passar a l'Espai Paper i definir el desplaçament en aquest sistema. Tenint en compte la correspondència dels punts **-0.30814,-0.44186** i **-1.19186,0.44186** de l'Espai Paper amb els homòlegs **8,0** i **0,8** de l'Espai Model, respectivament, un procediment seria usar coordenades d'Espai Model en **P-CURS**, partint de la posició **0,8**, calcular el factor d'escala per passar del segon al primer sistema, **0.110465**, i executar l'ordre **DESPLAZA** amb l'especificació del vector de desplaçament adaptat a l'Espai Paper. En comptes de la funció **M->P** que heu vist, comptaríem amb la funció **AP->NP**

```
  (defun AP->NP (AP NP) (* 0.110465 (- NP AP)))
```

i, amb unes diferències que s'han subratllat, **OMPLE-SUDOKU** començaria així

```
  (defun OMPLE-SUDOKU (/ CURSOR P-CURS SS C->F)
    (TECLAT)
    (if (= ABC "A")
        (progn
          (setq **9*9** (INI-COORDS) P-CURS '(0 8) POST T)
```

```
(INI-LLL)
(while (not (COMPLET-9*9 **9*9**))
       (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
       (TECLA-M)
       (command "ESPACIOP" "BORRA" SS ""
                "DESPLAZA" CURSOR ""
                           (mapcar 'AP->NP P-CURS (setq P-CURS P)) "")
       (MASCARA (ELEMENT LLL)) ...) ...) ...) ...)
```

L'altre procediment és el finalment adoptat, en què **P-CURS** es dóna en coordenades d'Espai Paper i allò que s'especifica a **"DESPLAZA"** són els punts inicial i final. En aquest cas, doncs, no només hem de parlar d'escalat sinó de translació, raó per la qual hem hagut de plantejar (amb els dos parelles de punts) i resoldre aquests dos sistemes d'equacions lineals

$$1)\quad \mathbf{P\_CURS_x = A \times P_x + B} \quad \begin{cases} \mathbf{-0.3081 = 8\,A + B} \\ \mathbf{-1.1919 = B} \end{cases}$$

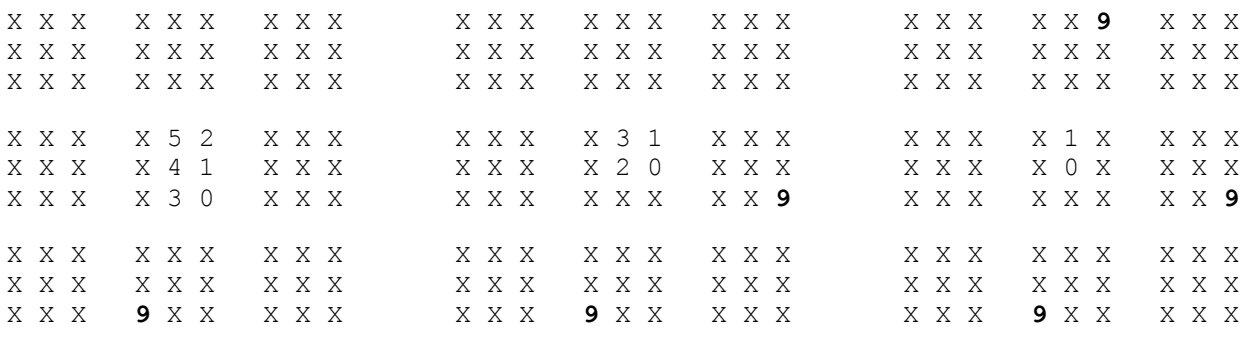
$$2)\quad \mathbf{P\_CURS_y = C \times P_y + D} \quad \begin{cases} \mathbf{-0.4419 = D} \\ \mathbf{0.4419 = 8\,C + D} \end{cases}$$

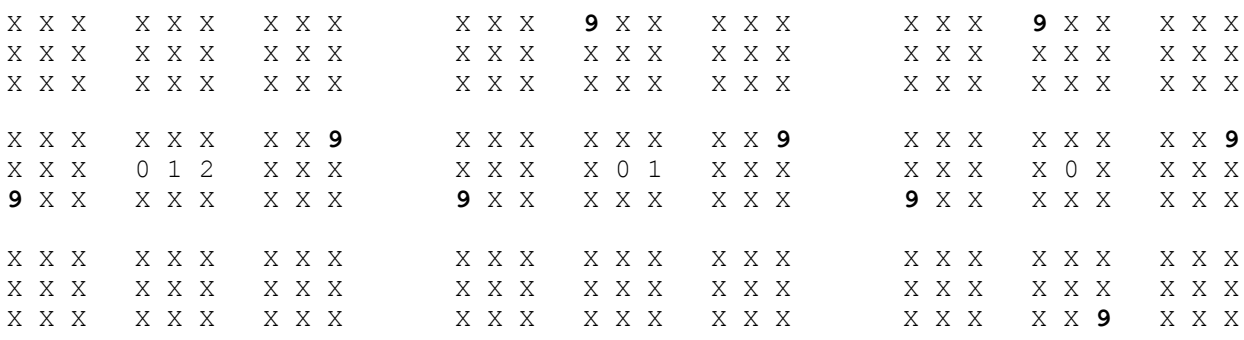
treient els valors **A** = **0.1105**, **B** = **-1.1919**, **C** = **0.1105** i **D** = **-0.4419**. Potser sense massa justificació pràctica (perquè de seguida veurem que amb l'opció A mai no ens caldrà moure el cursor 81 vegades, com amb la B) hem preferit aquest procediment a l'anterior, per por que una desfavorable acumulació d'errors de precisió en aquell (les noves posicions s'obtenien amb la suma vectorial de successius desplaçaments, mentre que en aquest d'ara cada vegada ens remetrem a dades de primera mà) pogués deixar el cursor ostensiblement mal col·locat sobre l'escaquer 9×9.

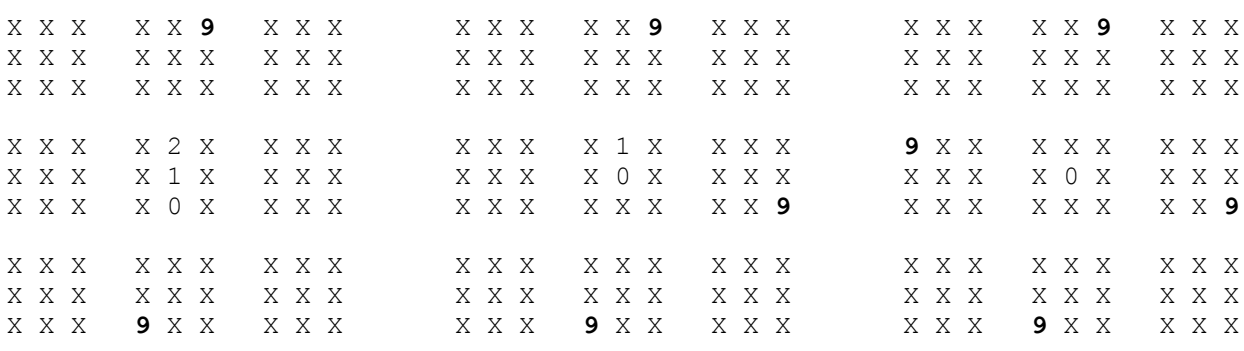
I això seria tot, si l'usuari estigués disposat a realitzar 81 assignacions i si l'expressió **(member N L)** de **MASCARA** (que, desfent substitucions, és equivalent a **(member N (ELEMENT LLL))** o a **(member N (nth X (nth Y LLL)))**, on **X** i **Y** representen les coordenades de **P**) fos condició necessària i suficient per garantir una SUDOKU-SOLUCIÓ, en lloc de ser només una de les diverses condicions necessàries. Però no. De la mateixa manera que la resolució manual d'un SUDOKU-PROBLEMA comença buscant i emplenant les caselles que només admeten un valor, amb l'exploració de requadres 9×3 (o 3×9), la localització dels valors que hi figuren dos cops i l'anàlisi del requadre 3×3 i, en concret, de la fila (columna) on hi és absent, en funció de les caselles plenes i de l'aparició del valor en el requadre concurrent 3×9 (o 9×3), la primera precaució mentre improvisem una solució serà tapar forats perillosos: esbrinar si després de l'última casella emplenada n'hi ha alguna de buida que ja només accepta un valor i assignar-l'hi automàticament, abans que se'ns acudeixi de fer-ho manualment i emplenar-la erròniament amb altres dels valors autoritzats per la màscara (que, mentre es limiti a presentar els valors trivialment compatibles amb el requadre 3×3 o amb la fila i columna concurrents, en general n'exhibirà més d'un). Des d'**ACTUALITZA** < **OMPLE-SUDOKU**, la funció **ACTUALITZA-PLUS** es farà càrrec d'aquesta missió, seguint certes pautes de cerca per detectar la presència d'algun dels 18 casos que oferim tot seguit. En aquestes 18 representacions esquemàtiques d'escaquer 9×9 parcialment emplenat suposarem que el valor de referència és **N** = **9**, 0 segueix significant que la casella és buida i X qualsevol possibilitat (que la casella és buida o que està ocupada per un valor compatible):

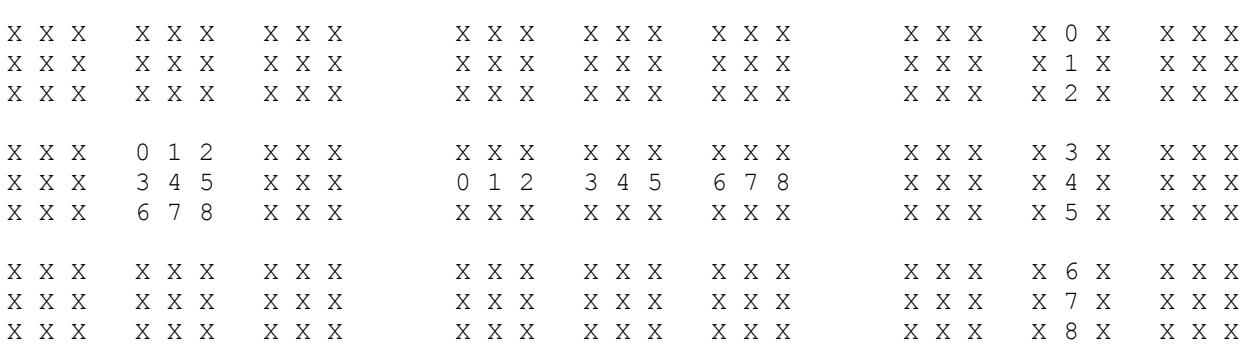
```
X X X   X X X   X X X      X X X   9 X X   X X X      X X X   9 X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X

X X X   0 1 2   X X X      X X X   X 0 1   X X X      X X X   X X X   X X 9
X X X   3 4 5   X X X      X X X   X 2 3   X X X      X X X   X 0 1   X X X
9 X X   X X X   X X X      9 X X   X X X   X X X      9 X X   X X X   X X X

X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
```

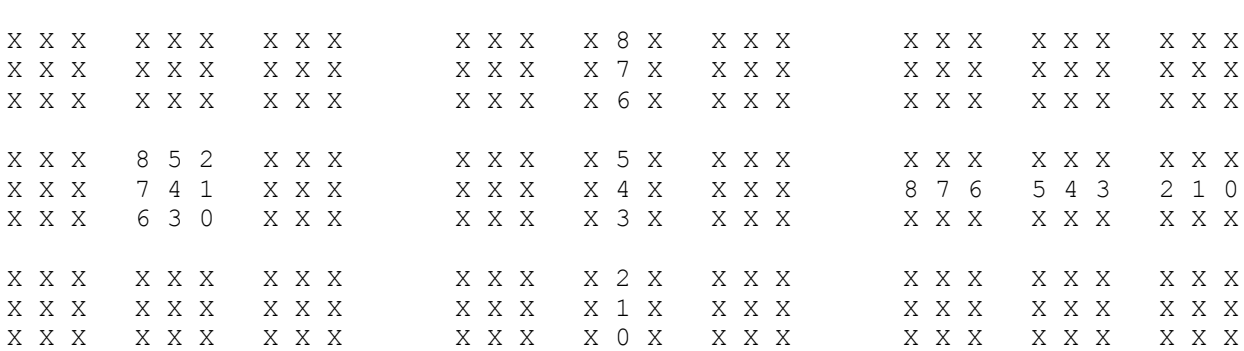
1-0 **1-1** **1-2**


1-0*
1-1*
1-2*


2-0
2-1
2-2


2-0*
2-1*
2-2*


0-0
F-1
C-1


0-0*
F-1*
C-1*

Si aquestes representacions les hem qualificat d'esquemàtiques no era només per la notació utilitzada en les caselles sinó perquè cada una no es refereix només a un cas concret sinó al conjunt definit per permutacions del tipus F1, F3, C1, C3 i PV (capítol precedent), del qual hem escollit com a icones el cas on en el requadre 3×3, fila o columna que aplega més valors explícits aquests apareixen ordenats, i el seu transposat. Aquesta intencionalitat genèrica s'ha intentat reforçar amb les denominacions que figuren al peu, que a més tracten de subsanar certa ambigüitat subjacent en l'ús de les X: així, **1-0** vol dir que podem considerar el requadre 3×3 que en l'exemple ocupa la posició central com intersecció d'un requadre 9×3 i un altre 3×9, i que el primer sols té una casella emplenada amb **9** i el segon no en té cap; **1-0*** es el transposat de **1-0**; **F-1**, vol dir fila amb només una casella buida, i **C-1**, columna amb només una casella buida. Resulta obvi que el conjunt **1-0*** està constituït pels transposats dels elements de **1-0**, que és en realitat el que volíem expressar en dir, per fer-ho més curt, que **1-0*** era el transposat de **1-0** (molt més enllà que la icona **1-0*** fos la transposada de la icona **1-0**), i a partir d'aquesta precisió, també podrem afirmar la identitat dels conjunts següents: **1-1 ≡ 1-1***, **2-2 ≡ 2-2***, **2-1 ≡ 1-2***, **1-2 ≡ 2-1***, **0-0 ≡ 0-0***, **F-1 ≡ C-1*** i **C-1 ≡ F-1***.

Però anem ja a la mare dels ous: com haureu vist de seguida, en tots aquests casos cal emplenar la casella 0 (l'única explícitament buida) amb el valor **N** = **9**, perquè no pot ser de cap altra manera. Així doncs, introduint la variable binària **PLUS**, que inicialment serà **nil** i que passarem a **T** en emplenar un forat, esquematitzarem així el procediment de cerca:

- Amb cadascun dels 3 requadres 9×3 de l'escaquer 9×9, si encara és **(not PLUS)**:
  - Amb cada valor 1... 9 representat per **N**, si encara és **(not PLUS)**:
    - Si el requadre té 1 o 2 caselles **N**:
      - Si té 1 casella **N**:
        - Amb cada un dels 2 requadres 3×3 que no tenen caselles **N**, si **(not PLUS)**:
          - Si el requadre 3×9 que hi incideix no té cap casella **N** o només en té 1 (si en té 2 és el cas **1-2 ≡ 2-1***, i resultarà més pràctic tractar-lo amb el segon):
            - Amb les 4 o 6 caselles del requadre 3×3 que no comparteixin fila ni columna amb cap **N**, compta les que estiguin buides.
            - Si només n'hi havia 1 (casos **1-0** i **1-1 ≡ 1-1***), emplena-la amb **N** i activa **PLUS**.
      - Si té 2 caselles **N**:
        - Amb el requadre 3×3 que no té cap casella **N**:
          - Si el requadre 3×9 que hi incideix té 2 caselles **N**:
            - Si l'única casella del requadre 3×3 que no comparteix fila o columna amb cap **N** és buida (cas **2-2 ≡ 2-2***), emplena-la amb **N** i activa **PLUS**.
          - Si no en té cap o només en té 1:
            - Amb les 2 o 3 caselles del requadre 3×3 que no comparteixin fila ni columna amb cap **N**, compta les que estiguin buides.
            - Si només n'hi havia 1 (casos **2-0** i **2-1 ≡ 1-2***), emplena-la amb **N** i activa **PLUS**.
- Si encara és **(not PLUS)**:
  - A cadascun dels 9 requadres 3×3 de l'escaquer, si encara és **(not PLUS)**:
    - Compta les caselles buides.
    - Si només n'hi havia 1:
      - Identifica l'únic valor **N** absent del requadre (cas **0-0 ≡ 0-0***), emplena la casella amb **N** i activa **PLUS**.
- Si encara és **(not PLUS)**:
  - A cadascuna dels 9 files de l'escaquer, si encara és **(not PLUS)**:
    - Compta les caselles buides.
    - Si només n'hi havia 1:
      - Identifica l'únic valor **N** absent de la fila (cas **F-1 ≡ C-1***), emplena la casella amb **N** i activa **PLUS**.
- Si encara és **(not PLUS)**:
  - Amb cadascun dels 3 requadres 3×9 de l'escaquer, si encara és **(not PLUS)**:
    - Amb cada valor 1... 9 representat per **N**, si encara és **(not PLUS)**:
      - Si el requadre té 1 o 2 caselles **N**:
        - Si té 1 casella **N**:
          - Amb cadascun dels 2 requadres 3×3 que no tenen caselles **N** (únicament tractarem els requadres incidents 9×3 que no en tinguin cap, perquè si en tenen 1 o 2 ja els haurem tractat abans, com a cas **1-1 ≡ 1-1*** o com a cas **2-1 ≡ 1-2***), si **(not PLUS)**:

            - Amb les 6 caselles del requadre 3×3 que no comparteixin fila ni columna amb la **N**, compta les que estiguin buides.
            - Si només n'hi havia 1 (casos **1-0***), emplena-la amb **N** i activa **PLUS**.
        - Si té 2 caselles **N**:
          - Amb el requadre 3×3 que no té cap casella **N** (únicament tractarem el requadre incident 9×3 que no en tingui cap o només en tingui 1, perquè si en té 2 ja l'haurem tractat abans, com a cas **2-2 ≡ 2-2***):
            - Amb les 2 o 3 caselles del requadre 3×3 que no comparteixin fila ni columna amb cap **N**, compta les que estiguin buides.
            - Si només n'hi havia 1 (casos **2-0*** i **1-2 ≡ 2-1***), emplena-la amb **N** i activa **PLUS**.
- Si encara és **(not PLUS)**:
  - A cadascuna dels 9 columnes de l'escaquer, si encara és **(not PLUS)**:
    - Compta les caselles buides.
    - Si només n'hi havia 1:
      - Identifica l'únic valor **N** absent de la columna (cas **C-1 ≡ F-1***), emplena la casella amb **N** i activa **PLUS**.

Quan toca passar a codi l'algorisme és fàcil adonar-se que, exceptuant la detecció del cas **0-0 ≡ 0-0***, en què n'hi ha prou a explorar els 9 requadres 3×3 un sol cop, l'anàlisi per requadres 9×3 (casos **1-0**, **1-1 ≡ 1-1***, **2-2 ≡ 2-2***, **2-0** i **2-1 ≡ 1-2***) i per requadres 3×9 (casos **1,0***, **2-0*** i **1-2 ≡ 2-1***) es podrien refondre en una formulació ambivalent que, introduint-hi els matisos que calgués, seria aplicada dues vegades, a l'igual dels rastrejos per files (cas **F-1 ≡ C-1***) i per columnes (cas **C-1 ≡ F-1***): només cal veure que en cadascuna d'aquestes situacions un dels dos subconjunts de caselles és el transposat de l'altre. És més: si apleguem tots tres rastrejos (per requadres 3×3, per requadres 9×3 o 3×9 i per files o columnes) en una mateixa seqüència que de primer executem amb la ****9*9**** original i després repetim amb la transposada, només hem de procurar que l'anàlisi per requadres 3×3 en sigui l'excepció i es faci únicament una vegada, en principi tant és la primera com la segona. Per què hem precisat "en principi"? Doncs perquè, si prescindim de la compactació que escometrem en passar a AutoLISP i linealitzem la seqüència, tal i com ho hem fet a la presentació en pseudocodi, tenim 10 possibilitats que no són indiferents a l'hora de considerar velocitat de processat:

 1) Requadres 3×3 + Files + Requadres 9×3 + Columnes + Requadres 3×9
 2) Requadres 3×3 + Requadres 9×3 + Files + Requadres 3×9 + Columnes
 3) Files + Requadres 3×3 + Requadres 9×3 + Columnes + Requadres 3×9
 4) Requadres 9×3 + Requadres 3×3 + Files + Requadres 3×9 + Columnes
 5) Files + Requadres 9×3 + Requadres 3×3 + Columnes + Requadres 3×9
 6) Requadres 9×3 + Files + Requadres 3×3 + Requadres 3×9 + Columnes
 7) Files + Requadres 9×3 + Columnes + Requadres 3×3 + Requadres 3×9
 8) Requadres 9×3 + Files + Requadres 3×9 + Requadres 3×3 + Columnes
 9) Files + Requadres 9×3 + Columnes + Requadres 3×9 + Requadres 3×3
10) Requadres 9×3 + Files + Requadres 3×9 + Columnes + Requadres 3×3

Abans d'abordar la problemàtica de la l'eficiència lligada a l'ordre dels diversos dispositius de detecció de forats, aclarirem que:
- L'esquema en pseudocodi precedent i el codi d'**ACTUALITZA-PLUS** que trobareu a la pàgina següent responen a l'ordenació 4.
- Com veureu en el codi esmentat, l'argument **ORIG** d'**ACTUALITZA-PLUS** val **T** en la 1ª volta i **nil** en la segona, reflectint les condicions en què s'executa la funció (si es processa la pseudomatriu ****9*9**** original o la seva transposada) per tal de regular l'aplicació de les excepcions. En relació a aquest tema, comprendreu que, si en comptes de l'ordenació 4 en volguéssiu una de diferent, caldria que introduíssiu alguns canvis:
  - Veureu que en les ordenacions de nombre parell (és el nostre cas), les files [columnes] van després dels requadres 9×3 [3×9], mentre que en les de nombre senar van abans. Així doncs, si volguéssiu la 2, 6, 8 o 10, **ACTUALITZA-PLUS** ja us va bé pel que fa a aquesta qüestió; però per canviar a l'ordenació 1, 3, 5, 7 o 9 caldría que l'exploració per files i columnes la traguéssiu del final i la poséssiu per davant de l'exploració de requadres 9×3 i 3×9.
  - Veureu que en 1 i 2 l'exploració de requadres 3×3 és la primera; en 3 i 4 es produeix entre la de files i la de requadres 9×3 (en la nostra, després dels requadres 9×3 i abans de les files); en la 5, després dels requadres 9×3 i abans de les columnes; en la 6, després de les files i abans dels requadres 3×9; en 7 i 8 es produeix entre l'exploració de columnes i la de requadres 3×9, i en la 9 i 10 ocupa l'última posició. Així doncs, si en volguéssiu una diferent de la 4 hauríeu de situar l'exploració de requadres 3×3 on calgués.

  - A més d'allò que s'indica en els dos subapartats precedents, si volguéssiu canviar a ordenació 7, 8, 9 o 10, on l'exploració de requadres 3×3 ja s'esdevé un cop hem deixat enrera la de files i requadres 9×3, caldria substituir la condició **(and ORIG (not PLUS))** per **(and (not ORIG) (not PLUS))** o l'equivalent **(not (or ORIG PLUS))**. Si voleu canviar a l'ordenació 1, 2, 3 o mantenir-vos en la 4 la condició existent, no s'ha de tocar (com a molt, en 1 o 2 ho podríeu simplificar deixant la condició en **ORIG**), i si voleu canviar a l'ordenació 5 o 6 resultarà indiferent mantenir-la o substituir-la.
- No tindria sentit decidir que és més eficient explorar les files abans [després] que els requadres 9×3, en processar la ****9*9**** original, i alhora que ho és més explorar les columnes després [abans] que els requadres 3×9, en processar-ne la transposada, i per això hem considerat 10 variants significatives i no pas 20.
- Tampoc hem considerat la possibilitat d'explorar columnes immediatament després de files i requadres 9×3 immediatament després dels 3×9 (o a l'inrevés), perque això suposaria haver de transposar ****9*9**** no una sinó dues (o tres) vegades.

Si ara pretenguéssim pronunciar-nos sobre la millor ordenació, hauríem de precisar en quines condicions definim "millor", perquè si ens referim a la major part de les avaluacions d'**ACTUALITZA**, corresponents a emplenaments manuals no seguits de cap emplenament automàtic a càrrec de **(ACTUALITZA-PLUS T)** + **(ACTUALITZA-PLUS ())**, l'ordre en què se succeeixin les diverses exploracions és indiferent, perquè totes hi desfilaran. Una altra cosa és quan la cerca localitza algun forat i aquí sí que hauríem d'establir un rànquing de rapidesa, ni que fos basant-nos exclusivament en la quantitat d'accessos a l'entretinguda funció auxiliar **SUB-X*Y**:

  1) 9 exploracions de files i 9 de columnes, a cap de les quals no s'hi accedeix.
  2) 9 exploracions de requadres 3×3, a cadascuna de les quals hi ha un accés.
  3) 3 exploracions de requadres 9×3 i 3 més de 3×9, a cadascuna de les quals li toquen 2 accessos, de mitjana, perquè a cada 3 exploracions hi ha 6.

Tot i que l'últim dispositiu es podria millorar, reduint de 12 a 6 els accessos a **SUB-X*Y**, seria molt discutible que això l'autoritzés a ocupar el segon lloc, i com a única veritat incontestable quedaria afirmar que el conjunt de les exploracions per files i columnes és força més ràpid que la resta. Això significa que 7 i 9 són les ordenacions més recomanables?: tampoc, com a norma aplicable a tot el procés d'emplenament, perquè només cap a les acaballes seria probable completar una fila, columna o requadre 3×3; mentrestant, de segur que l'exploració de requadres 9×3 o 3×9 és la més activa tapant forats i compensa situar-la abans, malgrat la durada. Sospesar el pro i el contra de cada opció i optimitzar els dispositius fins i tot donaria per a un *paper*, però no ens hi matarem més. Per què?: la resposta, a la fi del capítol i sobretot cap a la meitat del següent. De moment aquest codi servirà:

```
(defun 3-1 (I) (if (= I 0) '(1 2) (if (= I 1) '(0 2) '(0 1))))

(defun 3-2 (JK) (if (member 0 JK) (if (member 1 JK) 2 1) 0))

(defun SUB-X*Y (X*Y P1 P2 / J K L LL)
  (setq K -1)
  (foreach Y X*Y
    (setq K (1+ K) J -1)
    (if (not (or (< K (cadr P1)) (> K (cadr P2))))
        (progn
          (foreach X Y
            (setq J (1+ J))
            (if (not (or (< J (car P1)) (> J (car P2)))) (setq L (cons X L))))
          (setq LL (cons (reverse L) LL) L ()))))
  (reverse LL))

(defun E-MEMBER (N X*Y P / K E L)
  (setq K -1)
  (foreach Y X*Y
    (if (setq K (1+ K) E (member N Y))
        (setq L (cons (list (+ (car P) (- (length Y) (length E))) (+ (cadr P) K))
                      L))))
  (reverse L))

(defun PLUS-N->P ()
  (setq A-N N A-P K PLUS T)
  (if REAL (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))))

(defun NC<2 ()
```

```
  (if (= NL 2) (setq X2 (1- X1)))
  (setq X1 (if (= NC 1) (caar CC)) I 0)
  (foreach Y (if (= NL 1) 2Y (list Y1))
    (if (= NL 1) (setq X2 (1- X)))
    (repeat 3 (if (and (/= (setq X2 (1+ X2)) X1)
                       (listp (setq J (nth X2 (nth Y **9*9**)))))
                  (setq K J I (1+ I)))))
  (if (= I 1) (PLUS-N->P)))

(defun ACTUALITZA-PLUS (ORIG / M N CC NL NC 2X 2Y X1 X2 Y1 Y2)
  (foreach X1 '(0 3 6)
    (setq M (cons (SUB-X*Y A-9*9* (list X1 0) (list (+ X1 2) 8)) M)))
  (setq M (reverse M))
  (foreach Y1 '(0 3 6)
    (if (not PLUS)
        (progn
          (setq L (SUB-X*Y A-9*9* (list 0 Y1) (list 8 (+ Y1 2))))
          (foreach N L1A9
            (setq LL (E-MEMBER N L (list 0 Y1)) NL (length LL))
            (if (= NL 1)
                (progn
                  (setq 2X (mapcar '* (3-1 (/ (caar LL) 3)) '(3 3))
                        2Y (mapcar '+ (3-1 (rem (cadar LL) 3)) (list Y1 Y1)))
                  (foreach X 2X
                    (if (and (not PLUS)
                             (setq CC (E-MEMBER N (nth (/ X 3) M) (list X 0))
                                   NC (length CC))
                             (< NC 2)
                             (or ORIG (= NC 0)))
                        (NC<2))))
                (if (= NL 2)
                    (progn
                      (setq X1 (* (3-2 (list (/ (caar LL) 3) (/ (caadr LL) 3))) 3)
                            Y1 (+ Y1 (3-2 (list (rem (cadar LL) 3)
                                                (rem (cadadr LL) 3))))
                            CC (E-MEMBER N (nth (/ X1 3) M) (list X1 0))
                            NC (length CC))
                      (if (< NC 3)
                          (if (= NC 2)
                              (if ORIG
                                  (progn
                                    (setq X1 (+ X1 (3-2 (list (rem (caar CC) 3)
                                                              (rem (caadr CC) 3)))))
                                    (if (listp (setq K (nth X1 (nth Y1 A-9*9))))
                                        (PLUS-N->P))))
                              (NC<2)))))))))
  (if (and ORIG (not PLUS))
      (foreach Y1 '(0 3 6)
        (if (not PLUS)
            (foreach X1 '(0 3 6)
              (if (not PLUS)
                  (progn
                    (setq I 0 LL () CC (SUB-X*Y **9*9** (list X1 Y1)
                                                (list (+ X1 2) (+ Y1 2))))
                    (foreach C CC
                      (foreach J C
                        (if (listp J) (setq I (1+ I) K J) (setq LL (cons J LL)))))
                    (if (= I 1)
                        (foreach N L1A9
                          (if (not (or PLUS (member N LL))) (PLUS-N->P)))))))))
  (if (not PLUS)
      (foreach LL **9*9**
        (if (not PLUS)
            (progn
              (setq I 0)
              (foreach J LL (if (listp J) (setq I (1+ I) K J)))
              (if (= I 1)
                  (foreach N L1A9 (if (not (member N LL)) (PLUS-N->P)))))))))
```

Amb la implementació d'**ACTUALITZA-PLUS**, **ACTUALITZA** queda així:

```
(defun ACTUALITZA (REAL / L LL PLUS)
  (while (not PLUS)
         (setq LL () J -1)
         (foreach E **9*9**
           (setq J (1+ J) L (if (= J Y) (subst N P E) E) LL (cons L LL)))
         (setq **9*9** (reverse LL) LLL (ACT-LLL))
         (if REAL
             (progn
               (ACTUALITZA-PLUS T)
               (if (not PLUS)
                   (progn
                     (setq **9*9** (TRANSPOSAR **9*9**))
                     (ACTUALITZA-PLUS ())
                     (setq **9*9** (TRANSPOSAR **9*9**))))))
         (setq PLUS (not PLUS))))
```

Sembla que l'estructura d'**ACTUALITZA** se'ns ha complicat en relació a la versió que la precedia, però superada la primera impressió no costa massa d'apreciar que tot és una conseqüència lògica de la irrupció en escena d'**ACTUALITZA-PLUS**: l'argument **ORIG** de la funció no fa més que indicar si processa la pseudomatriu 9×9 original o la transposada (aquesta informació és necessària perquè en analitzar la casuística de les caselles amb valor predeterminat hem vist com uns casos eren més fàcils de copsar explorant per files, altres per columnes i n'hi havia algun d'ambivalent); d'altra banda, només sortirem d'**ACTUALITZA** després de l'actualització rutinària de ****9*9**** i **LLL** de resultes de l'emplenament manual de la casella **P** amb el valor **N**, quan ni **(ACTUALITZA-PLUS T)** ni **(ACTUALITZA-PLUS ())** hagin tapat cap forat perquè, si un dels dos ho hagués fet (**PLUS** = **T**), abans de passar un altra vegada el doble control per veure si l'emplenament d'un forat n'havia creat d'altres, haurem de reflectir la nova situació a ****9*9**** i a **LLL**. La manera més expeditiva (no diem elegant) de lligar-ho tot serà encabir-ho en un motor d'iteracions condicionades a **(not PLUS)**, raó per la qual caldrà tancar-lo invertint el valor assolit per **PLUS**.

Pel que fa a l'argument binari **REAL** d'**ACTUALITZA**, que condiciona la intervenció d'**ACTUALITZA-PLUS**, veiem que l'únic accés a **ACTUALIZA**, directe des d'**OMPLE-SUDOKU**, li assigna el valor **T**. ¿És que n'hi pot haver d'altres, executats des d'alguna funció interposada, en què **REAL** sigui **nil**? És clar que sí: si no, **REAL** no tindría cap raó de ser. Avancarem que això passa des d'una funció subordinada a **MASCARA** i anomenada **RE-MEMBER** (no la busqueu cap enrera, perquè l'última versió de **MASCARA** era provisional i encara no hi figurava aquesta funció), i avancem també per què en els seus accessos a **ACTUALITZA-PLUS** l'argument **REAL** serà **nil**: perquè **RE-MEMBER** és un dispositiu infal·lible per evitar emplenaments que ens barrin el pas a una SUDOKU-SOLUCIÓ, però és massa lent com per sobrecarregar els accessos obligats a **ACTUALITZA** amb **ACTUALITZA-PLUS** (que, per la seva infal·libilitat, no necessita). De fet, només amb el "tapaforats" **ACTUALITZA-PLUS** no estem protegits al cent per cent contra la introducció de valors inconvenients malgrat que **LLL** els autoritzi, ja que subsisteix un perill tan o més gran que l'existència de caselles buides amb un valor potencial predeterminat: el d'aquelles on no poden utilitzar-se tots els oferts per la màscara de **MASCARA**, perquè alguns durien inevitablement a un carreró sense sortida (una casella que ja no admet cap valor perquè tots són incompatibles amb els del seu requadre 3×3 o amb els de la fila o columna concurrent). En aquest cas l'acció preventiva no passa per assignar automàticament un valor a la casella, perquè n'hi ha de diversos, sinó per canviar la dinàmica de prospecció de valors: en comptes de presentar-nos tots els primàriament compatibles (informació que **LLL** manté permanentment actualitzada), cal que **MASCARA** en faci una tria, per mitjà de simulacions d'emplenament de la resta de caselles que ens donin la seguretat que amb cada valor seleccionat hi haurà almenys una SUDOKU-SOLUCIÓ. No ha d'estranyar que un rastreig tan exhaustiu de la viabilitat de cada candidat sigui un procés més aviat lent, i no us penseu que l'exclusió del dispositiu **ACTUALITZA-PLUS** dels accessos a **ACTUALITZA** efectuats des de **RE-MEMBER** n'és l'única conseqüència: això només ha estat el pretext que ens ha servit per referir-nos a **RE-MEMBER**, funció que encara trigarem a descriure en detall però de la qual calia començar a parlar, precisament perquè la seva lentitud condicionarà força el disseny d'**OMPLE-SUDOKU**. Igual com dintre de sis capítols (*Condicions mínimes*), quan ja es disposi d'una SOLUCIÓ, CREADA, COPIADA o CANÒNICA, i haguem començat tranquil·lament el procés d'activació de caselles (emplenament amb el valor predeterminat per la solució),

en constatar que ja es donen les condicions mínimes perquè el conjunt de caselles emplenades pugui ser un SUDOKU-PROBLEMA en tota regla s'activarà l'alarma **POST** per tal de realitzar l'inventari de les solucions compatibles, inventari que portarà el seu temps i que convé ajornar tot el que sigui possible, també a **OMPLE-SUDOKU** farem **POST** = **T** per caracteritzar el moment a partir del qual la presentació dels valors per omplir una casella ha d'anar més enllà d'una aplicació primària de les regles del joc.

Així que haurem de referir-nos a l'existència d'un abans i un després en la forma que té **MASCARA** d'assistir l'usuari mostrant-li els valors 1... 9 que pot assignar a cada casella, perquè si al començament els valors mostrats en el caseller 3×3 flueixen directament de l'expressió

```
(foreach N L1A9
  (if (member N (ELEMENT LLL))
      (command "TEXTO" ... (itoa N) ...)))
```

ultrapassada la fita **POST** que delimita les dues etapes del camí (una d'arcàdica, on n'hi ha prou a evitar valors que ja figurin a la fila, columna o requadre 3×3, i una altra, de més semblant a la travessa d'un camp de mines, on un pas en fals pot representar la fi), l'expressió passarà a ser

```
(foreach N L1A9
  (if (and (member N (ELEMENT LLL)) (RE-MEMBER **9*9** LLL))
      (command "TEXTO" ... (itoa N) ...)))
```

Quan activarem aquí l'alarma **POST**? Doncs quan es produeixi algun dels 6 supòsits que descrivirem tot seguit, els tres primers definits en requadres 9×3 o 3×9 i els tres últims que hi incorporen una columna o fila transversal:

1) Que en algun requadre 9×3 o 3×9 el nombre de caselles plenes sumat al nombre de valors diferents que hi surten arribi a **12**. Aquest supòsit, a més de respondre a situacions tan concretes com les d'aquests requadres 9×3

0 0 0	0 0 0	**7** 0 0	0 0 0	0 0 0	**7** 0 0	0 0 0	0 **7** 0	0 0 0
0 0 0	0 0 0	0 0 0	0 0 0	4 5 6	0 0 0	0 0 0	5 0 6	0 0 0
1 2 3	4 5 6	0 0 0	0 0 0	1 2 3	0 0 0	1 2 0	0 0 0	0 3 4

en què l'activació de **POST**, immediatament després d'haver emplenat sis caselles amb els valors 1... 6, impedirà que a la casella on veieu el **7** en negreta pugui realitzarse aquesta assignació (a la màscara ja no hi sortirà aquest valor, que tanmateix és compatible amb el requadre 3×3, amb la fila i suposem que amb la columna que ocupa), també servirà per limitar el marc general d'actuació i fer possible que la formulació dels supòsits següents sigui relativament senzilla.

2) Entrarem en el detall dels requadres 9×3 i les seves files, acceptant de forma implícita la vigència dels enunciats en la transposició als requadres 3×9 i les seves columnes. Si considerem els requadres 9×3 com una llista de tres files, cadascuna formada per tres tercets de caselles i usem els símbols **i**,**j**,**k** i **l**,**m**,**n** per representar valors 1, 2 o 3, el supòsit es pot esquematitzar així:

- Si en el **tercet-i**/**fila-l** totes les caselles estan plenes, amb valors **A**, **B**, **C**:
  - Si en el **tercet-j**/**fila-m** (**j** ≠ **i** i **m** ≠ **l**)hi ha alguna casella amb un valor **D** ≠ **A**, **B**, **C**:
    - Si en el **tercet-k**/**fila-n** (**k** ≠ **i**, **j** i **n** ≠ **l**, **m**) hi ha alguna casella buida:
      - Si en el **tercet-i/fila-n** i en el **tercet-k/fila-l** no hi ha cap casella amb el valor **D**:
        - Fes **POST** = **T**.

Del tipus de configuració a detectar, per impedir que sigui possible introduir valors com l'**1** en negreta, en mostrem aquests sis requadres 9×3

0 0 0	0 0 0	0 0 **1**	0 0 0	0 0 0	7 0 **1**	0 0 0	0 0 0	7 8 **1**
0 0 0	7 8 9	0 0 0	0 0 0	7 8 9	0 0 0	0 0 0	7 8 9	0 0 0
1 0 0	0 0 0	0 0 0	1 0 0	0 0 0	0 0 0	1 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 2 **1**	0 0 0	0 0 0	0 2 **1**	0 0 0	0 0 0	7 0 **1**
0 0 0	7 8 9	0 0 0	0 2 0	7 8 9	0 0 0	0 0 0	7 8 9	0 0 0
1 0 0	0 2 0	0 0 0	1 0 0	0 0 0	0 0 0	1 0 7	0 0 0	0 8 0

Adoneu-vos que la mare dels ous és un tercet ple (7-8-9 en tots sis exemples) que fa impossible la presència simultània d'un quart valor en els dos tercets que no en comparteixen fila ni columna (**1**, també en tots sis exemples).

3) Com en **2**, només entrarem en el detall dels requadres 9×3 i les seves files, els considerarem com una llista de tres files, cadascuna d'elles composta de tres tercets de caselles, però ara formarem tots els parells de tercets possibles en dues files (consecutius o no en la fila, però arrenglerats per columnes) i en compararem el contingut per detectar valors coincidents. La cosa anirà així:
- Si el **sextet-i+j**/**fila-l** té en comú amb el **sextet-i+j**/**fila-m** almenys 2 valors:
  - Si només són 2 valors **A**, **B**:
    - Si el **tercet-k**/**fila-n** (**k ≠ i**, **j** i **n ≠ l**, **m**) només té una casella plena, amb un valor **C ≠ A**, **B**:
      - Fes **POST** = **T**.
  - Altrament (hi ha 3 valors comuns **A**, **B**, **C**):
    - Si el **tercet-k**/**fila-n** (**k ≠ i**, **j** i **n ≠ l**, **m**) és buit o només té una casella plena, amb un dels valors **A**, **B**, **C**:
      - Fes **POST** = **T**.

En primer lloc, convé aclarir que els dos sextets no pot tenir 4 valors comuns perquè dues files de 4 valors (diferents), com els d'aquests dos requadres 9×3,

```
4 0 3   0 0 0   2 0 1       0 0 4   0 0 0   3 2 1
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
1 0 2   0 0 0   3 0 4       1 2 3   0 0 0   4 0 0
```

amb un total de 8 caselles plenes amb 4 valors diferents, ja no hauríen passat del **supòsit 1**. Dels tipus de configuració a detectar, perquè no sigui possible introduir valors com el **9** en negreta, en mostrem aquests sis requadres 9×3, dos de cadascun dels casos contemplats

```
0 0 0   0 0 0   0 2 1       0 0 2   0 0 0   0 0 1       0 0 0   0 0 0   3 2 1
0 0 0   8 9 0   0 0 0       0 0 0   8 9 0   0 0 0       0 0 0   0 9 0   0 0 0
1 2 0   0 0 0   0 0 0       1 0 0   0 0 0   2 0 0       1 2 3   0 0 0   0 0 0


3 0 0   0 0 0   0 2 1       3 0 0   0 0 0   0 2 1       0 0 0   0 0 0   3 2 1
0 0 0   0 9 0   0 0 0       0 0 0   1 9 0   0 0 0       0 0 0   1 9 0   0 0 0
1 2 0   0 0 0   0 0 3       1 2 0   0 0 0   0 0 3       1 2 3   0 0 0   0 0 0
```

Aquí la mare dels ous és el tercet que no comparteix fila ni columna amb els dos sextets que tenen 2 o 3 valors comuns (1-2 en els dos primers exemples, o 1-2-3 en els demés), en què emplenar una nova casella amb un valor diferent d'aquests (**9**, en tots ells) faria impossible incloure'ls tots, com és obligat.

4) Només entrarem en el detall dels requadres 9×3 i les seves files, considerant-los com una llista de tres files, cadascuna d'elles composta de tres tercets de caselles, però igual que en **3** formarem tots els parells de tercets possibles en dues files (consecutius o no en la fila, però arrenglerats per columnes) i en compararem el contingut per detectar valors coincidents. Ara, però, caldrà anar més enllà del requadre 9×3 per veure si la intersecció 3×3 amb algun requadre 3×9 disposa únicament de una o dues caselles per assignar-hi un o dos valors que no poden desplaçar-se a cap altre posició pel fet d'aparèixer en dues de les files del requadre 9×3 i en una de les columnes del requadre 3×9. Ho podreu seguir millor sobre aquests tres casos (X són caselles buides o compatibles):

```
X X X   X X 1   X X X       X X X   X X 1   X X X       X X X   X X 1   X X X
X X X   X X X   X X X       X X X   X X 2   X X X       X X X   X X 2   X X X
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X X X   X X X

1 X X   X X X   X X X       1 2 X   X X X   X X X       1 2 X   X X X   X X X
X X X   X 1 X   X X X       X X X   1 2 X   X X X       X X X   0 3 X   X X X
X X X   X X X   X X 1       X X X   X X X   X 1 2       X X X   X X X   X 1 2

X X X   X X X   X X X       X X X   X X X   X X X       X X X   X X X   X X X
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X X X   X X X
X X X   1 X X   X X X       X X X   X X X   X X X       X X X   X X X   X X X
```

A l'esquerra tenim un requadre 9×3 on dues de les files tenen el valor 1, i el requadre 3×9 que el talla justament en el quadrat 3×3 desprovist d'aquest valor té dues columnes també amb el valor 1. Això comporta que sols en una de les nou caselles d'aquest quadrat (la casella central, a l'exemple) s'hi pot ubicar un 1, però en aquest cas no hi haurà cap perill que ningú assigni un altre valor a

la casella, fent que el requadre 3×3 quedi privat del seu 1 i el sudoku sense solució, perquè així que apleguem els 4 valors 1 perifèrics, **ACTUALITZA-PLUS** hi posarà d'ofici l'**1** que apareix en negreta. L'exemple del mig ja s'acosta més al que ens ocupa, però la presència de un 1 en una de les dues posicions possibles permet un altre cop que **ACTUALITZA-PLUS** hi intervingui assignant un **2** a l'altra casella i salvant la situació. Però a la dreta no tindríem tanta sort: les dues files amb els valors 1 i 2 i la columna amb aquests mateixos valors sols deixen en el requadre 3×3 dues caselles (4,5 i 5,5) aptes per rebre'ls; si res no ho impedís, la introducció d'un valor diferent en alguna d'aquestes caselles (el **3** en negreta, per exemple) seria fatal. Esquematitzem així l'algorisme salvador:

- Si el **sextet-i+j**/**fila-l** té 2 valors en comú amb el **sextet-i+j**/**fila-m** (si en tingués més s'activaria el **supòsit 3**):
  - Fes una llista **F12** amb aquests valors comuns.
  - Amb cadascuna de les 3 columnes que passen pel **tercet-k**/**fila-n** (**k ≠ i, j** i **n ≠ l, m**):
    - Fes una llista **L** amb els valors de la columna, tret dels que pertanyen al requadre 9×3 de partença.
    - Si els 2 valors de la llista **F12** coincideixen amb els de la llista **L**:
      - Si les 2 caselles del **tercet-k**/**fila-n** alienes a la columna no estan ocupades (si ho estan ha de ser amb els valors comuns a **F12** i **L**, perquè si n'hi hagués algun de diferent s'hauria activat el **supòsit 3**):
        - Fes **POST** = **T**.

Algú podria argumentar que, encara que les dues darreres caselles esmentades estiguessin buides, si en columna amb una d'elles hi hagués un dels dos valors comuns (imagineu-vos que, en l'exemple del mig, l'**1** situat a 4,5 el baixem fins a 4,1), ja ens hauríem ficat de peus a la galleda perquè seria massa tard per activar **POST**. Doncs no: tant en aquest cas com en el d'haver posat prèviament un **2** al costat de l'**1** (a 5,1), **ACTUALITZA-PLUS** haurà emplenat les caselles amb els valors adients.

5) Entrarem en el detall de les tres files de cadascun dels requadres 9×3 i de les nou columnes, acceptant de forma implícita que tot el que diem és transposable a les tres columnes de cada requadre 3×9 i les nou files. Suposem que, en tots aquests 3 escaquers 9×9 hem començat emplenant les caselles 1,4 (1) i 4,5 (2).

```
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X 1 X   X X X      X X X   X 1 X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X

X X X   X X X   1 X X      X X X   X X X   1 X X      X X X   X X X   1 X X
X X X   2 1 3   X X X      X X X   2 0 1   X X X      X X X   2 0 3   X X X
1 X X   X X X   X X X      1 X X   X X X   X X X      1 X X   X X X   X X X

X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
```

Seguint amb 6,5 (3) i 7,6 (1), tant és l'ordre amb què ho fem, **ACTUALITZA-PLUS** emplenarà automàticament el forat 5,5 amb **1** (a l'esquerra), i això impedirà que a la columna 5ª pugui entrar-hi cap altre 1. Si, en lloc de seguir amb 6,5 (3), emplenem 7,6 (1) i 5,8 (1), tant és l'ordre amb què ho fem, el forat que taparà amb un **1 ACTUALITZA-PLUS** serà 6,5 (centre). Però si, abans d'emplenar 7,6 (1), seguim amb 6,5 (3) i 5,8 (1), tant és l'ordre amb què ho fem, haurem begut oli perquè res no ens impedirà situar a la casella 7,6 un altre **1** (en negreta, a la dreta), entrant en un carreró sense sortida: a la fila 5ª ja no s'hi pot posar el valor 1. És en previsió de situacions com aquesta que caldrà implementar un nou supòsit (també aplicat a cada requadres 9×3, pero ara amb la intervenció de columnes senceres), segons l'esquema:

- Si en el **tercet-i**/**fila-l** només hi ha una casella buida:
  - Amb cadascun dels valors **A** de la columna a què pertany la casella esmentada (exclosos els que hi pugui haver en el requadre):
    - Si a la **fila-l** no hi ha cap valor **A**:
      - Si la **fila-m** té el valor **A** però la **fila-n** no (**m, n ≠ l**):
        - Fes **POST** = **T**.
      - Altrament:
        - Si la **fila-n** té el valor **A** però la **fila-m** no:
          - Fes **POST** = **T**.

6) Com en **5**, només entrarem en el detall de les tres files de cada requadres 9×3 i de les nou columnes. Ara caldrà detectar un requadre 3×3 amb només 4 caselles plenes que formin un rectangle (és a dir, que cada una se situï a la fila d'una altra i a la columna d'una tercera) i, en les dues alineacions deixades pel quartet (una fila i una columna, buides en el requadre 3×3), caldrà verificar si hi ha algun valor diferent que aparegui en una però no a l'altra: el perill rau en què l'usuari ompli amb aquest valor alguna casella lliure de la segona, perquè llavors el requadre 3×3 en quedaría privat. Donem per fet que aquestes alineacions han de tenir caselles lliures fora del requadre, perquè tenint-ne entre dues i quatre de plenes (segons difereixin o no dels valors d'aquest) ja anem a parar al **supòsit 1**. D'altra banda, si el requadre tingués més de quatre caselles plenes no caldria preocupar-se perquè, o bé el cinquè valor coincideix amb el de l'alineació, i aleshores desapareix el risc, o bé que es diferent, i aleshores la suma de les caselles plenes i dels valors diferents dóna 12, i de nou el **supòsit 1** activa **POST**. Menys evident és que calgui quadrar el rectangle per crear la situació de perill: no podría passar res omplint només 3 caselles? Doncs no, i ho veurem en els dos primers escaquers. A l'esquerra, si emplenem 2,1 (1), 4,2 (2), 4,3 (4), 6,2 (3) i 5,8 (1), s'activarà **POST** per aplicació del **supòsit 5** i **ACTUALITZA-PLUS** posarà automàticament un 1 a la casella 6,3, cosa que pot passar de dues formes: amb l'ordre d'emplenament descrit, almenys pel que fa a les dues últimes caselles emplenades manualment, i en aquest cas **POST** s'activa a conseqüència d'haver posat un 1 a la columna 5ª que aporta el buit al tercet de la fila 2ª, completat pels valors 2 i 3, tot interactuant amb la fila 1ª que també té un 1, i aquí hi intervé **ACTUALITZA-PLUS** que tapa el forat 6,3 abans que l'usuari pugui emplenar una nova casella; emplenant 5,8 (1) quan la casella 4,3 encara és buida, i en aquest cas **POST** s'activa a conseqüència de tenir un 1 la fila 1ª que aporta el buit al tercet de la columna 4ª, completat pels valors 2 i 4, tot interactuant amb la columna 5ª que té un altre 1, i ara és quan l'usuari intervé introduint-hi el 3 (que, malgrat **POST**, figura entre els valors autoritzats), fet seguit de la intervenció d'**ACTUALITZA-PLUS**. Davant d'això, a algú se li pot ocórrer d'emplenar la casella 6,5 amb un 1, abans de fer-ho a 5,8, per tal d'impedir la intervenció d'**ACTUALITZA-PLUS**, com es mostra en el centre: el resultat serà avançar l'aplicació del **supòsit 5**, que ara es produirà immediatament després d'emplenar la casella 6,5 (passa com a l'últim cas, llevat que la fila 1ª no interactuarà amb la columna 5ª sinó amb la 6ª), de manera que el valor 1 ja no serà candidat a l'emplenament de la casella 5,8. Això és precisament el que en casos com el de la dreta caldria aconseguir: que, un cop tinguem el requadre 9×3 inferior com el veieu, amb el 3×3 central ocupat pels valors 2, 3, 4 i 5 disposats en rectangle, ja no puguem introduir cap més valor sense un control estricte, perquè, més enllà del requadre esmentat, un 1 a la columna 5ª liquidaria la viabilitat del sudoku.

```
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X 1 X   X X X      X X X   X(1)X   X X X      X X X   X 1 X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X

X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X 1   X X X      X X X   X X X   X X X
X X X   X X X   X X X      X X X   X X X   X X X      X X X   X X X   X X X

X X X   4 0 1   X X X      X X X   4 0 0   X X X      X X X   4 0 5   X X X
X X X   2 0 3   X X X      X X X   2 0 3   X X X      X X X   2 0 3   X X X
X 1 X   0 0 0   X X X      X 1 X   0 0 0   X X X      X 1 X   0 0 0   X X X
```

Seguint amb el tercer escaquer, adoneu-vos que en el requadre 3×3 inferior dret no hi pot haver cap 1. Això, que és obvi en el tercet inferior, també ho és per als altres dos: si algun tingués aquest valor, **ACTUALITZA-PLUS** n'afegiria un altre en el requadre inferior central (entre el 2 i el 3, o entre el 4 i el 5), i després d'això s'activaria **POST**, per ser d'aplicació el **supòsit 1** (7+5 = 12). Esquematizant com en els 4 casos precedents, i deixant clar que no cal repetir amb la matriu transposada si l'original no aconsegueix activar **POST**, tenim:

- Si en el **tercet-i**/**fila-l** només hi ha una casella buida:
  - Si en el **tercet-i**/**fila-m** només hi ha una casella buida (**m ≠ l**):
    - Si la casella pertany a la mateixa columna que la de **tercet-i**/**fila-l**:
      - Fes una llista **V** amb els quatre valors.
      - Fes una llista **L** dels valors de les caselles plenes de la columna que passa per la casella buida.
      - Fes **COND** = **A**.

- Altrament:
  - Si en el **tercet-i**/**fila-m** les tres caselles són buides:
    - Fes una llista **L12** dels valors de les caselles plenes de la **fila-m**.
    - Fes **COND** = **B**.
- Si **COND = A, B**:
  - Si **COND** = **A**:
    - Si en el **tercet-i**/**fila-n** les tres caselles són buides (**n ≠ l, m**):
      - Fes una llista **L12** dels valors de les caselles plenes de la **fila-n**.
      - Fes **COND** = **C**.
  - Altrament (**COND** = **B**):
    - Si en el **tercet-i**/**fila-n** només hi ha una casella buida:
      - Si la casella pertany a la mateixa columna que la de **tercet-i**/**fila-l**:
        - Fes una llista **V** amb els quatre valors.
        - Fes una llista **L** dels valors de les caselles plenes de la columna que passa per la casella buida.
        - Fes **COND** = **C**.
  - Si **COND** = **C**:
    - Fes una llista **L12** ajuntant el contingut de **L** i **L12**.
    - Amb cadascun dels valors de la llista **L12**:
      - Si el valor no coincideix amb cap dels quatre de la llista **V**:
        - Fes **POST** = **T**.

En relació a la penúltima línia de pseudocodi, aclarirem que la llista **L12** (que en l'exemple aplega els valors ubicats a la fila 1ª i a la columna 5ª) pot ser buida i pot tenir valors repetits, sempre que també figurin en el rectangle; però si n'hi ha algun que no hi figuri (aquest activarà **POST**) ha de ser únic: si de primer entrem els dos valors 1 i després pretenem fer el rectangle, el **supòsit 5** intervindrà si es comença dibuixant-ne un costat (dos valors, a la mateixa fila o columna) o serà **ACTUALITZA-PLUS** qui el tanqui amb un 1 si s'ha començat amb una diagonal (tres valors), tot plegat respostes com les descrites en l'escaquer de l'esquerra; si el rectangle no estava complet quan hem entrat el primer valor diferent, tornarà a ser el **supòsit 5** el que activarà **POST**, com també havíem vist (esquerra i centre). Però el **supòsit 6** ha estat expressament concebut per introduir el primer valor amb el rectangle ja acabat.

Grosso modo, els sis supòsits es distribueixen entre diverses funcions, sense que la correspondència sigui exacta i sense cap pretensió d'haver lograt l'articulació òptima. Més realista seria admetre que, entre moltes provatures, la necessitat de més supòsits a partir d'una hipòtesi que l'autor va arrossegar durant força temps (en escriure aquestes línies, tot just fa mes i mig que els únics supòsits en joc eren els que ara anomenem **1** i **5**) i la seva correcta formulació van anar sorgint de forma successiva i en ocasions imprevisible, gairebé sempre a conseqüència de la descoberta d'una configuració no contemplada pels precedents. I aquesta progressió ha comportat reestructuracions dràstiques però també sargits i pedaços per adaptar les novetats amb el mínim enrenou. Així que, a partir de l'accés des d'**ACTUALITZA**,

```
(if (not POST) (T+D<12))
```

la funció principal **T+D<12** ha pres el nom del **supòsit 1** (nombre **T**otal de caselles plenes **+** nombre de caselles **D**iferents **< 12**), l'aplica i després té un doble accés (matriu original i transposada) a la funció **DETECTA**, que al seu torn s'ocupa dels **supòsits 5** i **6**, imbricat el segon en el primer per aprofitar les dades parcialment compartides, i que finalment accedeix a **ABC** (**supòsit 2**) i **AB/BA** (**supòsits 3** i **4**):

```
(defun ABC/D (/ F1 F2 F3 F/3 I J1 J2 J3 K1 K2)
  (foreach K1 '(0 1 2)
    (setq F1 (nth K1 3*F))
    (foreach J1 '(0 1 2)
      (setq F/3 (nth J1 F1))
      (if (and (not POST) (= (length F/3) 3))
          (foreach K2 (3-1 K1)
            (setq F2 (nth K2 3*F)
                  F3 (nth (3-2 (list K1 K2)) 3*F))
            (foreach J2 (3-1 J1)
              (setq J3 (3-2 (list J1 J2)))
              (foreach I (nth J2 F2)
                (if (and (not (member I F/3))
                         (not (member I (nth J3 F1)))
                         (not (member I (nth J1 F3)))
                         (< (length (nth J3 F3)) 3))
                    (setq POST T)))))))))
```

```
(defun AB/AB (/ F1 F2 F/3 F12 FF1 FF2 I J1 J2 J3 K1 K2)
  (foreach E COMB32
    (if (not POST)
        (progn
          (setq K1 (car E) K2 (cadr E)
                F1 (nth K1 3*F) F2 (nth K2 3*F))
          (foreach F COMB32
            (if (not POST)
                (progn
                  (setq J1 (car F) J2 (cadr F) F12 ()
                        FF1 (append (nth J1 F1) (nth J2 F1))
                        FF2 (append (nth J1 F2) (nth J2 F2)))
                  (if (and (> (length FF1) 1) (> (length FF2) 1))
                      (progn
                        (foreach G FF1
                          (if (member G FF2) (setq F12 (cons G F12))))
                        (if (or (= (setq I (length F12)) 2) (= I 3))
                            (progn
                              (setq J (3-2 F)
                                    F/3 (nth J (nth (3-2 E) 3*F))
                                    J3 (if (= (length F/3) 1) (car F/3)))
                              (if (or (and (= I 2) J3 (not (member J3 F12)))
                                      (and (= I 3)
                                           (or (not F/3) J3 (member J3 F12))))
                                  (setq POST T))))
                        (if (and (not POST) (> (length F12) 1))
                            (progn
                              (setq X (1- (* 3 J))
                                    Y (if (= Y0 -1)
                                          '(2 5)
                                          (if (= Y0 2) '(-1 5) '(-1 2))))
                              (repeat 3
                                (if (not POST)
                                    (progn
                                      (setq X (1+ X) I 0 L ())
                                      (foreach G Y
                                        (repeat 3
                                          (setq G (1+ G)
                                                L (cons (nth X (nth G 9*9)) L))))
                                      (foreach G F12
                                        (if (member G L) (setq I (1+ I))))
                                      (if (> I 1)
                                          (progn
                                            (setq I 0 F/3 (nth (3-2 F)
                                                               (nth (3-2 E) 3*F)))
                                            (foreach G F12
                                              (if (member G F/3) (setq I (1+ I))))
                                            (if (< I 2)
                                                (setq POST T))))))))))))))))))
```

(Aclarim que a **OMPLE-SUDOKU** haurem fet **(setq COMB32 (list '(0 1) '(0 2) '(1 2)))**, i aquesta llista és un artifici perquè els parells d'índexs **K1** i **K2**, **J1** i **J2** vagin adoptant els valors corresponents a totes les combinacions binàries de **0**, **1** i **2**.) Ja veieu que aquesta versió d'**AB/AB** és un trasto, pel que fa al corriment excessiu cap a la dreta, fruit d'un ús massa canònic de dissuasions **(if (not POST) ...)**. Si les reduïm, al preu d'unes poques iteracions inútils, molt breus, queda més curta:

```
(defun AB/AB (/ F1 F2 F/3 F12 FF1 FF2 I J1 J2 J3 K1 K2)
  (foreach E COMB32
    (if (not POST)
        (setq K1 (car E) K2 (cadr E)
              F1 (nth K1 3*F) F2 (nth K2 3*F)))
    (foreach F COMB32
      (if (not POST)
          (setq J1 (car F) J2 (cadr F) F12 ()
                FF1 (append (nth J1 F1) (nth J2 F1))
                FF2 (append (nth J1 F2) (nth J2 F2))))
```

```
      (if (and (not POST) (> (length FF1) 1) (> (length FF2) 1))
          (progn
            (foreach G FF1 (if (member G FF2) (setq F12 (cons G F12))))
            (if (or (= (setq I (length F12)) 2) (= I 3))
                (progn
                  (setq J (3-2 F) F/3 (nth J (nth (3-2 E) 3*F))
                        J3 (if (= (length F/3) 1) (car F/3)))
                  (if (or (and (= I 2) J3 (not (member J3 F12)))
                          (and (= I 3) (or (not F/3) J3 (member J3 F12))))
                      (setq POST T))))
            (if (and (not POST) (> (length F12) 1))
                (progn
                  (setq X (1- (* 3 J))
                        Y (if (= Y0 -1) '(2 5) (if (= Y0 2) '(-1 5) '(-1 2))))
                  (repeat 3
                    (if (not POST)
                        (progn
                          (setq X (1+ X) I 0 L ())
                          (foreach G Y
                            (repeat 3
                              (setq G (1+ G) L (cons (nth X (nth G 9*9)) L))))
                          (foreach G F12
                            (if (member G L) (setq I (1+ I))))
                          (if (> I 1)
                              (progn
                                (setq I 0 F/3 (nth (3-2 F) (nth (3-2 E) 3*F)))
                                (foreach G F12
                                  (if (member G F/3) (setq I (1+ I))))
                                (if (< I 2) (setq POST T)))))))))))))))

(defun DETECTA (ORIG / 9*9 X Y X0 Y0 C0 C1 C2 L0 L1 L2 L12 V W V1 V2 3*F F)
  (setq 9*9 (if ORIG **9*9** (TRANSPOSAR **9*9**)))
  (foreach Y0 '(-1 2 5)
    (setq Y Y0 3*F ())
    (if (not POST)
        (progn
          (repeat 3
            (setq Y (1+ Y) L0 (nth Y 9*9) F ())
            (foreach X0 '(-1 2 5)
              (setq X X0 J 0 V ())
              (if (not POST)
                  (progn
                    (repeat 3
                      (setq X (1+ X) K (nth X L0))
                      (if (listp K) (setq J (1+ J) C0 X) (setq V (cons K V))))
                    (setq F (cons V F))
                    (if (= J 1)
                        (progn
                          (setq J -1 L ())
                          (repeat 9
                            (setq J (1+ J))
                            (if (or (<= J Y0) (> J (+ 3 Y0)))
                                (setq K (nth C0 (nth J 9*9))
                                      L (if (atom K) (cons K L) L))))
                          (setq K (3-1 (- Y Y0 1))
                                L1 (nth (+ Y0 1 (car K)) 9*9)
                                L2 (nth (+ Y0 1 (cadr K)) 9*9)
                                X (+ X0 4) J 0 V1 () V2 ())
                          (repeat 3
                            (setq X (1- X) V1 (cons (nth X L1) V1)
                                           V2 (cons (nth X L2) V2)))
                          (if ORIG
                              (progn
                                (setq J 0 L12 () W ())
                                (foreach E V1
                                  (if (listp E)
                                      (setq J (1+ J) C1 (car E))
                                      (setq W (cons E W))))
```

```
                                (if (or (and (= J 1) (= C1 C0)) (= J 3))
                                    (progn
                                      (if (= J 3)
                                          (setq L12 L1)
                                          (setq V (append V W)))
                                      (setq J 0 W ())
                                      (foreach E V2
                                        (if (listp E)
                                            (setq J (1+ J) C2 (car E))
                                            (setq W (cons E W))))
                                      (if (or (and (not L12) (= J 3))
                                              (and L12 (= J 1) (= C2 C0)))
                                          (progn
                                            (if (not L12)
                                                (setq L12 L2)
                                                (setq V (append V W)))
                                            (setq W ())
                                            (foreach E L12
                                              (if (atom E) (setq W (cons E W))))
                                            (foreach E (append W L)
                                              (if (not (or POST (member E V)))
                                                  (setq POST T)))))))))
                            (if (not POST)
                                (foreach E L
                                  (if (and (not (or POST (member E L0)
                                                    (member E V1) (member E V2)))
                                           (or (and (member E L1)
                                                    (not (member E L2)))
                                               (and (member E L2)
                                                    (not (member E L1)))))
                                      (setq POST T)))))))))
              (setq 3*F (cons (reverse F) 3*F))) ; Com que la disposició de files
            (if (not POST) (ABC/D))              ; de 3*F és indiferent, no cal fer
            (if (not POST) (AB/AB))))))          ; (setq 3*F (reverse 3*F))

(defun T+D<12 ()
  (foreach LL (list **9*9** (TRANSPOSAR **9*9**))
    (if (not POST)
        (foreach Y '(-1 2 5)
          (if (not POST)
              (progn
                (setq J 0 K ())
                (repeat 3
                  (setq Y (1+ Y))
                  (foreach X (nth Y LL)
                    (if (atom X)
                        (setq J (1+ J) K (if (member X K) K (cons X K))))))
                (if (>= (+ J (length K)) 12) (setq POST T)))))))
  (if (not POST) (DETECTA T))
  (if (not POST) (DETECTA ())))
```

Potser **DETECTA** l'hauríem hagut d'anomenar **DETECTA-1**, substituir el codi

```
                            (if ORIG
                                (progn
                                  ....................................
                                  (if (or (and (= J 1) (= C1 C0)) (= J 3))
                                      (progn
                                        ................................
                                        (if (or (and (not L12) (= J 3))
                                                (and L12 (= J 1) (= C2 C0)))
                                            (progn
                                              ............................
                                              (foreach E (append W L)
                                                (if (not (or POST (member E V)))
                                                    (setq POST T)))))))))
```

per una senzilla crida **(DETECTA-2)**
i traslladar-lo a la definició d'aquesta nova funció: així, a banda la incomoditat de llegir expressions tan llargues sense gairebé cap més suport que la progressió

cap a la dreta per visualitzar-ne la jerarquia, hauríem diferenciat millor el codi que correspon al **supòsit 5** (**DETECTA-1**) del que s'ocupa del **supòsit 6** (**DETECTA-2**). Però no ho hem fet, a diferència de les funcions **ABC** i **AB/BA**, en considerar que la imbricació instrumental del **supòsit 6** en el **5** serà de més bon seguir: fixeu-vos en què l'exploració de l'escaquer 9×9 avança per requadres 9×3, a cadascun d'ells per files, a cada fila per tercets i que únicament ampliarà el camp de visió quan en trobi un amb dues caselles plenes i una de buida; ho farà comptant amb la columna que passa pel buit i amb les altres dues files que componen el requadre 9×3, però mentre que amb això ja n'hi ha prou per determinar si l'aplicació del **supòsit 5** és pertinent, pronunciar-se sobre el **6** exigeix més verificacions en el requadre 3×3.

Observeu també que, en una dinàmica general en què les comprovacions corresponents a cada supòsit es realitzen sobre la transposada de la pseudomatriu **A-9*9** (si **POST** no s'ha activat) després d'haver-les aplicat a l'original, el **6** n'és una excepció, com havíem anunciat: la disposició rectangular de les úniques 4 caselles plenes i la ambivalència de la fila i columna concurrents fa innecessària la repetició. Per cert que, si no s'ha unificat sota un mateix cicle l'aplicació d'aquesta dinàmica, i el **supòsit 1** s'aplica als tres requadres 9×3 i als tres requadres 3×9 abans de passar a la resta de supòsits, no ha estat només amb la intenció de rendibilitzar el filtratge més dràstic i menys complicat, sinó per garantir la cobertura del **6**.

En relació a l'esmentat **supòsit 6**, acabarem amb una precisió per justificar la no exacta correspondència entre l'esquema de funcionament proposat d'entrada i la traducció concreta a codi avaluable. Entre les pàgines 135/136, l'esquema indicava que, fins haver localitzat en el mateix requadre 3×3 dos tercets amb un sol buit cada un i situats ambdós a la mateixa columna (**tercet-i**/**fila-l** i **tercet-i**/**fila-m**), no creàvem la llista **L** dels valors de les caselles plenes d'aquesta columna, i així él risc d'haver-se pres la molèstia inútilment es reduïa raonablement: com a molt, podia esdevenir-se que el **tercet-i**/**fila-n** no fos del tot buit. Tanmateix, si passem al codi de **DETECTA**, veurem que **L** es construeix així que s'ha localitzat el primer, **tercet-i**/**fila-l**. En el context que presentàvem línies enrera, l'explicació és simple: l'esquema pretenia il·lustrar el **supòsit 6** aïlladament, mentre que la funció **DETECTA** reuneix els **supòsits 5** i **6** (i després dóna pas a **2**, **3** i **4**), evitant reiteracions en aquells processos que poden ser compartits; i en aquest sentit és lògic que, havent-se acomplert la condició principal d'identificació del cas **5**, ja es preparin les llistes **L**, **L1** i **L2**, també necessaris per identificar, si s'escau, el cas **6**, perquè si no es d'aplicació un supòsit pot ser-ho l'altre.

Amb la inclusió de l'accés a **T+D<12**, **ACTUALITZA** ja està complet. Però encara no us en mostrarem la versió definitiva: abans cal veure com la nova situació **POST** = **T** repercuteix en **MASCARA** i tenir perfilat l'intrument **RE-MEMBER** de cerca fina dels valors compatibles, a fi i efecte de presentar el codi conjuntament i tenir ocasió de comentar algunes afectacions mútues d'ordre pràctic, entre les quals hi ha la qüestió de la subordinació d'**ACTUALITZA-PLUS** a l'argument **REAL**.

L'esquema de funcionament de **MASCARA** (donem aquest nom a la funció en el sentit de matriu 3×3, composta de caràcters numèrics o blancs, que en cada moment s'aplica sobre el caseller dibuixat per **TECLAT**) seria:

MASCARA:
- Amb cadascun dels valors 1... 9:
  - Si, en primera instància, la casella admet el valor:
    - Fes **COMPLET** = **nil**
    - Si **POST** = **nil** o **RE-MEMBER** = **T**, incorpora'l a la llista de valors admissibles

Pel que fa a **RE-MEMBER** (el nom d'aquesta funció d'usuari vol expressar una segona volta sobre el resultat de la funció predefinida **member**, en la determinació dels valors que una casella buida pot admetre, no pas la traducció literal "recordar"), seria una funció recursiva definida així:

RE-MEMBER:
- Si **COMPLET** = **nil**:
  - Si l'escaquer 9×9 està complet, fes **COMPLET** = **T**
  - Si no:
    - Cerca la primera casella buida
    - Amb cadascun dels valors que, en primera instància, admeti aquesta casella:
      - Emplena-la
      - Avalua **RE-MEMBER**
- **COMPLET**

RE-MEMBER es va autorecorrent, tot baixant un nivell de recursió en passar de cada casella buida (que omple provisionalment) a la següent. Quan arriba a una casella buida (un determinat nivell de recursió) en què cap valor 1... 9 no és admissible, es retrocedeix per les caselles emplenades provisionalment (es remunten nivells de recursió) fins trobar-ne alguna on encara quedin valors admissibles per verificar (i es reinicia el procés, utilitzant el primer dels valors encara no assajats) o, si ja no en queden, fins retornar a la primera casella buida (on havíem començat). Així que, una de dues:
- En alguna d'aquestes incursions s'omple provisionalment l'última casella buida, s'esdevé que **COMPLET** = **T** i amb aquest valor es remunten tots els nivells.
- Havent arribat a una casella buida en què cap valor 1... 9 no és admissible, en retrocedir les caselles emplenades provisionalment no se'n troba cap on encara quedin valors per verificar i s'arriba al nivell de partença amb **COMPLET** = **nil**.

Per aprofitar la variable binària **OK** i evitar confusions amb la funció **COMPLET-9*9** (que, com veurem, incorporarà un argument) usarem **OK** en comptes de **COMPLET**.

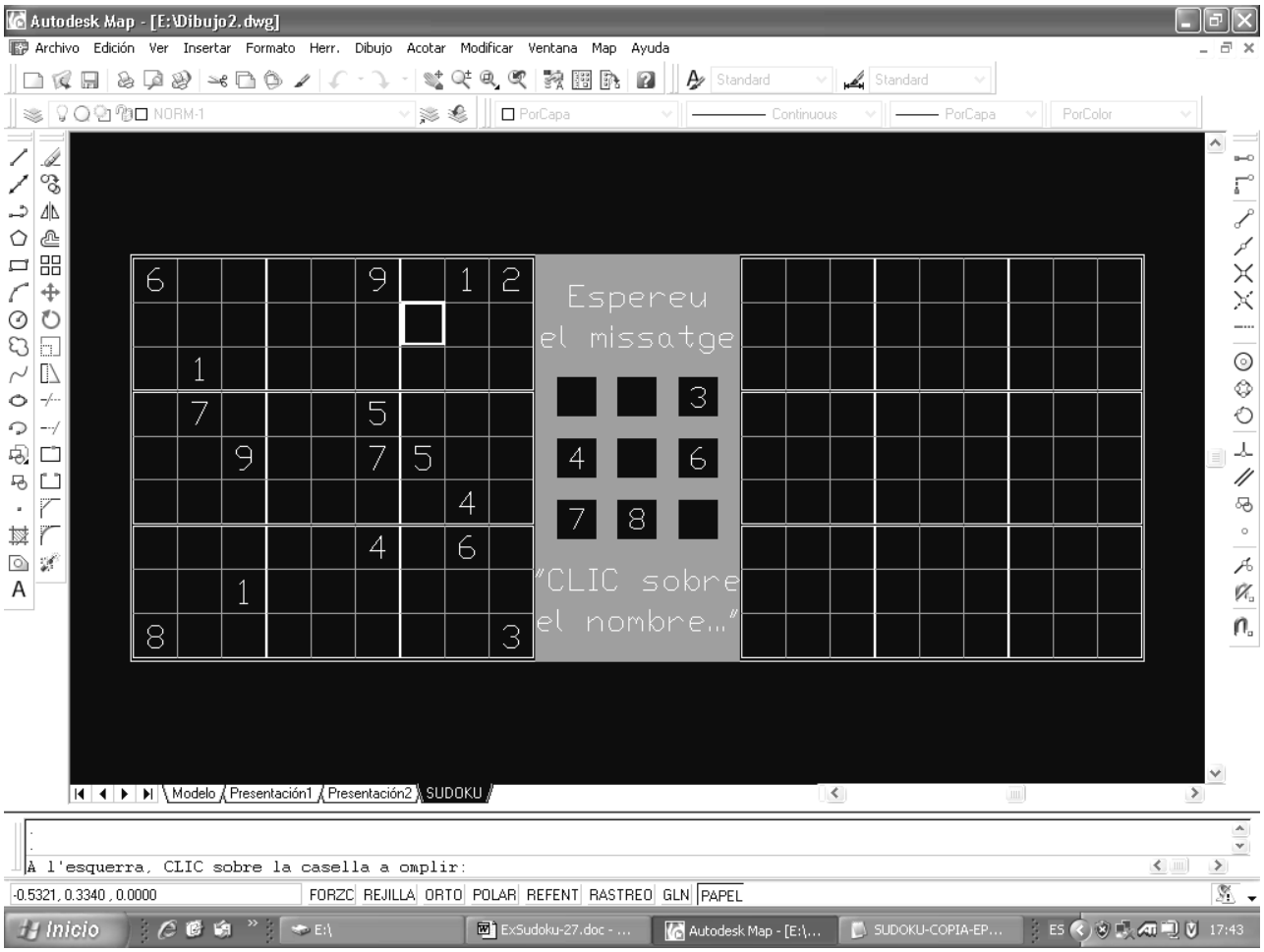
Tal i com ha quedat **ACTUALITZA**, el tapaforats **ACTUALITZA-PLUS**, d'ús obligat quan **POST** = **nil** (per evitar sudokus sense solució) i optatiu quan **POST** = **T** (per mirar de sostreure a **RE-MEMBER** algunes caselles i accelerar el procés), s'aplica sempre però només sobre els valors realment assignats a les caselles (**REAL** = **T**), perquè per fer-ho també quan **RE-MEMBER** busca la viabilitat d'aquests valors (**REAL** = **nil**) no seria rendible: introduir **ACTUALITZA-PLUS** a cada recursió encara alentiria el procés. De moment, la prudència ens aconsella deixar **POST** activada indefinidament, així que **T+D<12** detecti la primera situació de perill (més endavant ja revisarem aquesta decisió, si de cas), de manera que considerarem dividit el procés a càrrec d'**OMPLE-SUDOKU** (opció A) en dues etapes, com veníem anunciant: la prèvia a aquesta intervenció decisiva de **T+D<12** (**POST** = **nil**) i la posterior (**POST** = **T**).

Mentre l'escaquer 9×9 no estigui ple (**(not (COMPLET-9*9 **9*9**))**, en la primera etapa només s'accedirà a **ACTUALITZA** després d'haver fet clic en alguna casella (**(TECLA-M)**) i, entre els valors admissibles (**(MASCARA (setq LL (ELEMENT LLL)))**), d'haver-ne triat un (**(TECLA-P)**), i el resultat serà l'actualització de ****9*9**** i **LLL** (**(setq LLL (ACTUALITZA **9*9** N P LLL T) **9*9** (car LLL) LLL (cadr LLL))**). En la segona etapa, a més d'aquest accès n'hi hauran altres de secundaris des de **MASCARA** (més concretament, des de **(RE-MEMBER 0-**9*9** 0-LLL)**) que serveixen per fer possible l'exploració exhaustiva i no té efectes directes sobre ****9*9**** i **LLL**. Si ens limitem als accessos principals i tenim en compte allò que caracteritza les dues etapes, l'única diferència és que en la primera executem **T+D<12** per veure si s'escau activar **POST**, mentre que a la segona ja prescindim d'aquesta funció perquè mantenim **POST** = **T**, de manera que **ACTUALITZA** sempre realitza aquestes dues accions: actualitzar les pseudo-matrius locals **A-9*9** i **A-LLL**, que després seràn assignades a ****9*9**** i a **LLL** respectivament (si no s'actualitzen directament aquestes és per l'ús que es fa d'**ACTUALITZA** des de la funció autorecursiva **RE-MEMBER**); mirar amb **ACTUALITZA-PLUS** si hi ha algun forat per tapar (recordeu que parlem dels accessos principals a **ACTUALITZA**, perquè unes línies més amunt dèiem que en els secundaris des de **RE-MEMBER** < **MASCARA** en prescindiríem) i fer-ho si s'escau. I, com hem vist abans, cal encabir aquestes dues accions en l'estructura **(while (not PLUS) ...)** precisament en atenció als casos en què s'escaigui tapar forats. Doncs bé, resulta evident que en la primera etapa cal integrar **T+D<12** en aquesta estructura i ho ha de fer un cop **A-9*9** i **A-LLL** estiguin actualitzats, però ¿situem aquest detector de situacions crítiques abans o després d'**ACTUALITZA-PLUS**?: si no volem complicar el codi amb una segona actualització d'**A-9*9** i **A-LLL** quan aquesta funció hagi arribat a tapar algun forat i **PLUS** = **T** (recurs que seria contradictòria amb el dispositiu **(while (not PLUS) ...)**), l'hem d' incorporar abans, perquè aquesta és la manera de garantir que **T+D<12** analitzarà la situació després de l'última casella emplenada manualment i també després de tots els emplenaments automàtics que **ACTUALITZA-PLUS** hagi pogut desencadenar:

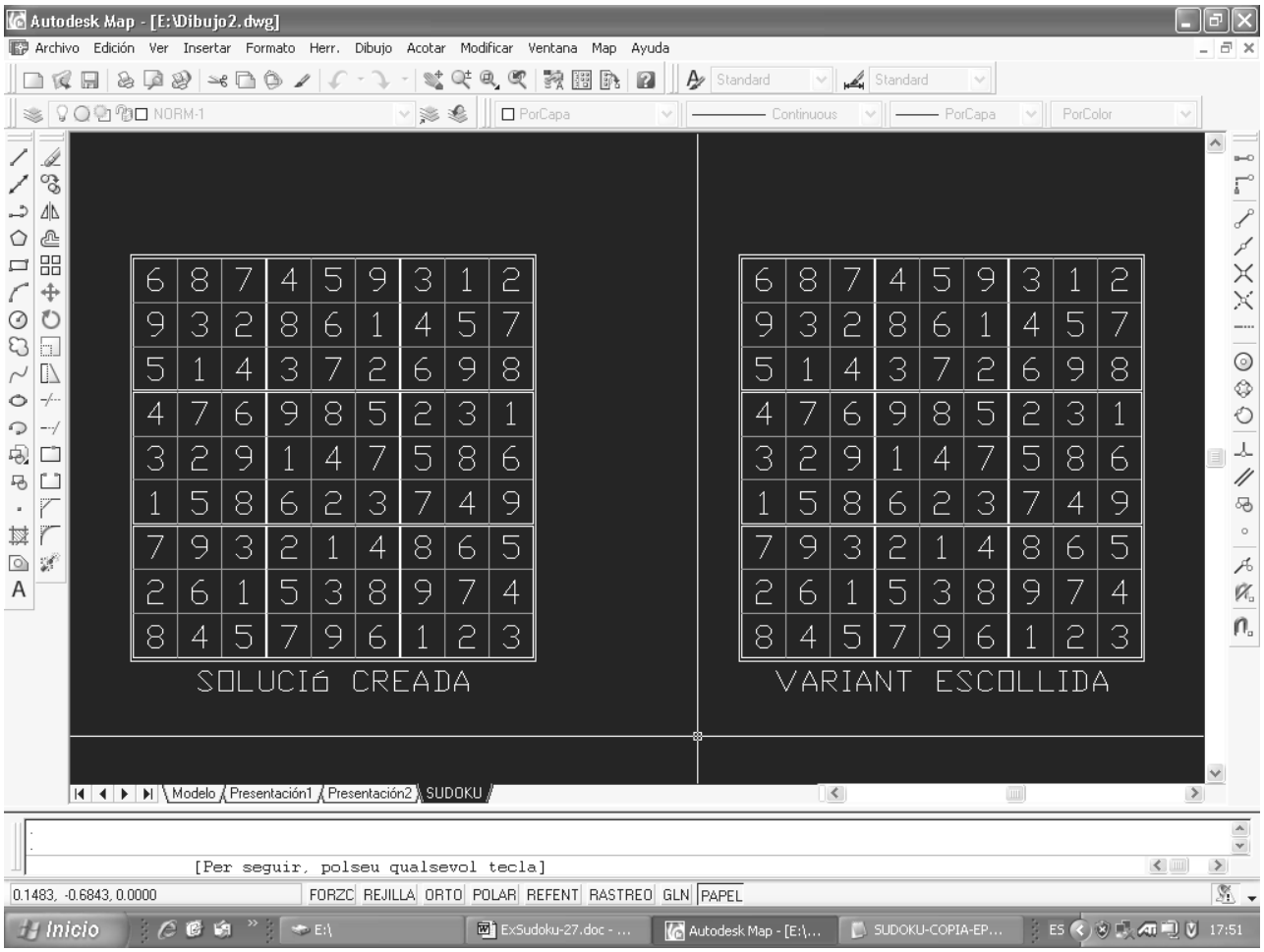
```
(defun ACT-LLL (X Y A-LLL / X/3 Y/3 LL L J I)
  (setq X/3 (/ X 3) Y/3 (/ Y 3) J -1)
  (foreach F A-LLL
    (setq L () I -1 J (1+ J))
    (foreach E F
      (setq I (1+ I)
            L (cons (if (= J Y)
                        (if (= I X) () (subst () A-N E))
                        (if (= I X)
                            (subst () A-N E)
```

```
                              (if (and (= (/ I 3) X/3) (= (/ J 3) Y/3))
                                  (subst () A-N E) E)))
                      L)))
    (setq LL (cons (reverse L) LL)))
  (reverse LL))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J)
                 L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST) (T+D<12))
         (if REAL
             (progn
               (ACTUALITZA-PLUS T)
               (if (not PLUS)
                   (progn
                     (setq A-9*9 (TRANSPOSAR A-9*9))
                     (ACTUALITZA-PLUS ())
                     (setq A-9*9 (TRANSPOSAR A-9*9))))))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))

(defun COMPLET-9*9 (9*9 / COMP)
  (setq COMP T)
  (foreach L 9*9
    (if COMP (foreach E L
               (if (and COMP (listp E)) (setq COMP () R-P (if POST E))))))
  COMP)

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)

(defun COL->FILA (/ N)
  (setq C->F (mapcar '(lambda (Y LL)
                        (setq Y (1- (* 3 (/ Y 3))) N 0)
                        (repeat 3
                          (setq Y (1+ Y))
                          (foreach E (nth Y LL) (if (atom E) (setq N (1+ N)))))
                      N)
                     (reverse P) (list **9*9** (TRANSPOSAR **9*9**))))
  (setq C->F (< (car C->F) (cadr C->F))))

(defun TRANSPOSA-HO ()
  (setq **9*9** (TRANSPOSAR **9*9**)
        LLL (TRANSPOSAR LLL) P (reverse P) 9**9 ())
  (foreach EE **9*9**
    (setq 0-L ())
    (foreach E EE (setq 0-L (cons (if (atom E) E (reverse E)) 0-L)))
    (setq 9**9 (cons (reverse 0-L) 9**9)))
  (setq **9*9** (reverse 9**9) K X X Y Y K))

(defun TREU-RETOL () (repeat 4 (command "BORRA" "LT" "")))
```

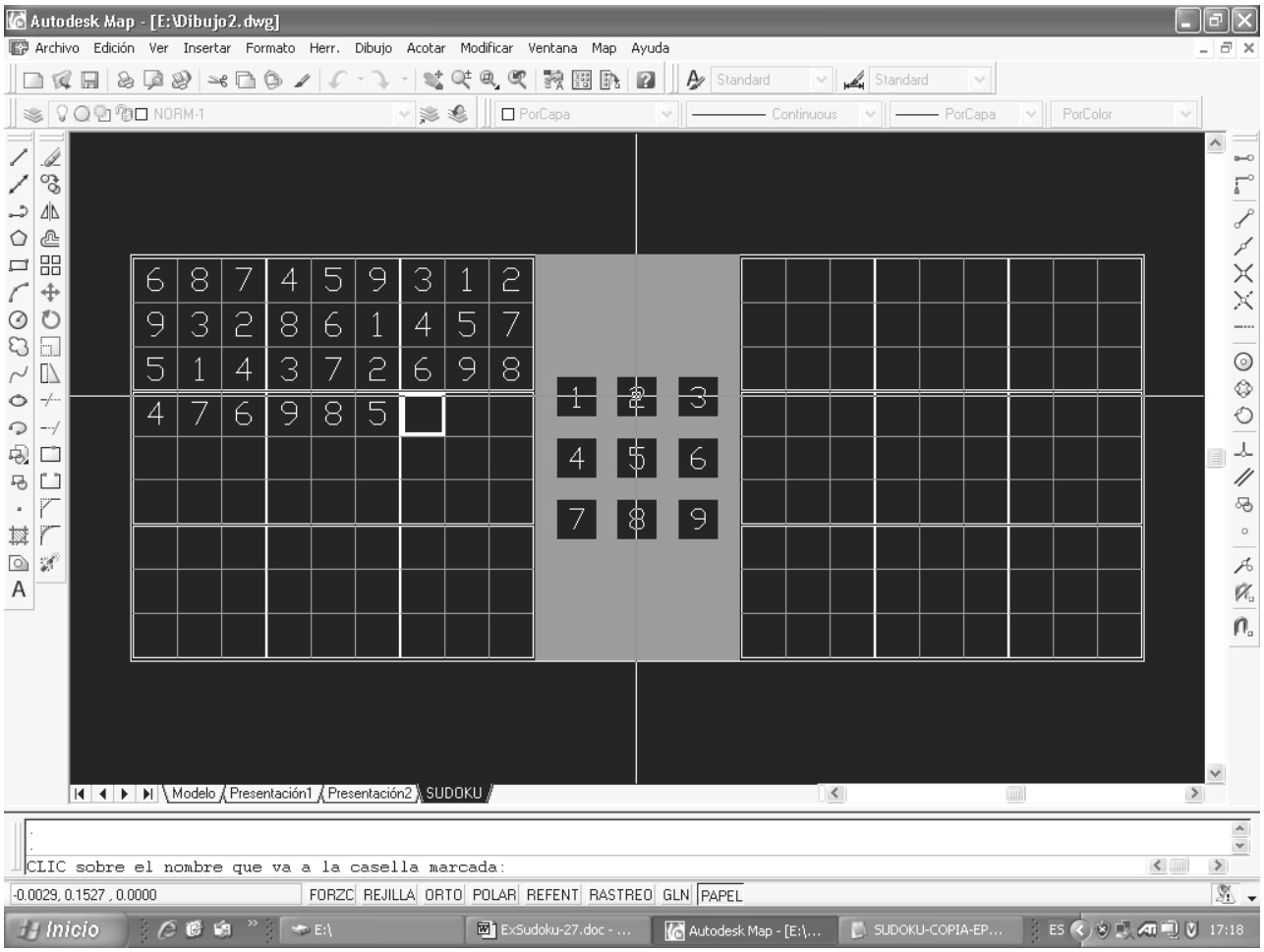
```
(defun MASCARA (L / INI OK JJ 9**9 0-**9*9** 0-LLL 0-L)
  (setq LL () SS (ssadd))
  (if POST
      (progn
        (if (COL->FILA) (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** **9*9** 0-LLL LLL)
            (progn
              (setq JJ (if (< Y 6) '(1 0 2) '(2 1 0))
                    0-LLL (F=3 **9*9**) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**)
                    0-LLL (F=3 LLL) P (list X (rem Y 3)))))))
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "CLIC sobre"
                      "el nombre..."))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (if POST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         ((= N 2)     '(0 0.15))
                         ((= N 3)  '(0.15 0.15))
                         ((= N 4) '(-0.15 0))
                         ((= N 5)     '(0 0))
                         ((= N 6)  '(0.15 0))
                         ((= N 7) '(-0.15 -0.15))
                         ((= N 8)     '(0 -0.15))
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if POST (TREU-RETOL))))
  (if POST
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))

(defun OMPLE-SUDOKU (/ COMB32 CURSOR P-CURS SS)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq COMB32 (list '(0 1) '(0 2) '(1 2))
              **9*9** (INI-COORDS)
              P-CURS '(-1.1919 0.4419))
        (INI-LLL)
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP" "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (TECLA-P ())
               (command "ESPACIOM" "CVPORT" 2
                        "CAPA" "D" "NORM-1" "" "TEXTO" "MC" P 0.5 0 (itoa N)
                        "CVPORT" 3
                        "CAPA" "D" "VAR-1"  "" "TEXTO" "MC" P 0.5 0 (itoa N)
                        "CVPORT" 2)
               (setq LLL (ACTUALITZA **9*9** N P LLL T)
                     **9*9** (car LLL) LLL (cadr LLL)))
        (command "ESPACIOP"))
      (progn ...)
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))
```



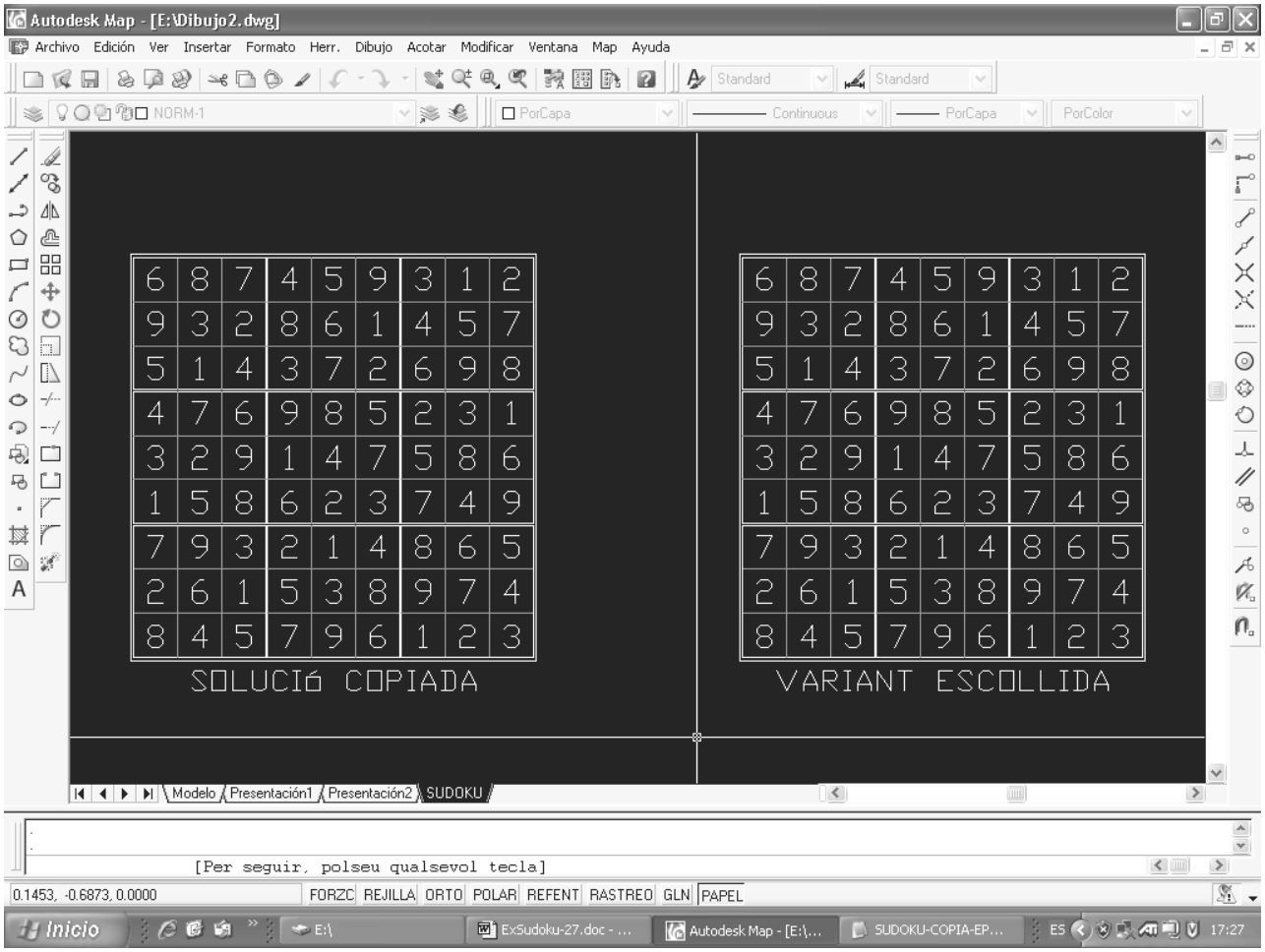
L'opcio A serveix per crear una SUDOKU-SOLUCIÓ (o, alternativament a l'opció B,...



per obtenir la SUDOKU-SOLUCIÓ copiant un SUDOKU-PROBLEMA i algunes caselles més).



L'opcio B serveix per copiar una SUDOKU-SOLUCIÓ (en aquest cas, la solució al...



sudoku difícil de LA VANGUARDIA de 25-02-08, usat també per il·lustrar l'opció A).

Hem cregut procedent repetir **COMPLETA-9*9** i la primera meitat d'**OMPLE-SUDOKU** (que és la que ara ens interessa) perquè, en ser accedides **COMPLETA-9*9** i **ACTUALITZA** també des dels diversos nivells de recursió de **RE-MEMBER**, no hi podíem gestionar directament les variables globals ****9*9****, **LLL**, **N** i **P**, que (conservant els noms en **OMPLE-SUDOKU** i canviant-los en **RE-MEMBER**) hem hagut de convertir en arguments: només la primera a **COMPLETA-9*9** i totes quatre (afegides a **REAL**) en **ACTUALITZA**. Això ens ha obligat a l'artifici de formar una llista aplegant-ne els dos primers, convertir-la en valor de la funció i recuperar ****9*9**** i **LLL** actualitzats tal i com es veu a les línies 4ª i 5ª començant pel final. Abans de presentar el codi ja ho havíem comentat, com de passada, però semblava de justícia insistir-hi. També hi ha coses a dir sobre **MASCARA** (parlant amb propietat, sobre alguns additaments d'aquesta funció, dels quals no s'ha dit res), però ho farem més endavant perquè, per anar amb un mínim d'ordre, abans hauríem d'aturar-nos de nou en **ACTUALITZA**. D'aquesta funció hem justificat la posició relativa d'**ACTUALITZA-PLUS** i **T+D<12**, i fins i tot l'existència de la variable **REAL**, però sobre ambdues funcions caldrà filar una mica més prim. Si retornem al **supòsit 4** i recuperem els tres exemples, al del mig no li passarà com al de l'esquerra (per al qual no havíem previst cap supòsit específic perquè **ACTUALITZA-PLUS** ja tapava amb un **1** el forat 4,4), sinó que **T+D<12** activarà **POST** i de seguida **ACTUALITZA-PLUS** emplenarà amb un **2** el forat 4,4. Però **POST** = **T** haurà provocat que **MASCARA** segueixi treballant amb **RE-MEMBER**, quan la situació que havia fet saltar el **supòsit 4** no era realment crítica: la casella 4,4 estava virtualment emplenada amb l'únic valor possible, i n'hauríem pogut emplenar unes quantes més sense que **MASCARA** necessités la participació de **RE-MEMBER** per realitzar exploracions tan feixugues com innecessàries. Així doncs, ¿hem de reconsiderar la conveniència que **T+D<12** actui abans que **ACTUALITZA-PLUS**? No, però sí que podríem revocar l'activació de **POST** quan detectéssim que el mateix emplenament manual que ha provocat aquest fet en desencadena un altre d'automàtic amb el qual l'activació ja no s'hauria produït. Només caldrà que una nova variable **NOU** ens informi que **ACTUALITZA-PLUS** ha actuat en la mateixa iteració **while** en què **T+D<12** havia activat **POST**:

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS NOU E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J)
                 L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST) (progn (T+D<12) (setq NOU POST)))
         (if REAL
             (progn
               (ACTUALITZA-PLUS T)
               (if (not PLUS)
                   (progn
                     (setq A-9*9 (TRANSPOSAR A-9*9))
                     (ACTUALITZA-PLUS ())
                     (setq A-9*9 (TRANSPOSAR A-9*9))))
               (if (and PLUS NOU) (setq POST () NOU ())))))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))
```

Queda clar que l'adopció d'una variable **A-9*9** local a **ACTUALITZA**, substituint la variable global ****9*9**** també haurà afectat a **T+D<12**, a la seva funció auxiliar **DETECTA** i a **ACTUALITZA-PLUS**. Però, així com en el cas de les dues primeres no ens molestarem a reproduir-les senceres, perquè la substitució sols afecta la segona línia, i n'hi haurà prou a presentar-vos els inicis,

```
(defun DETECTA (ORIG / 9*9 X Y X0 Y0 C0 C1 C2 L0 L1 L2 L12 V W V1 V2 3*F F)
  (setq 9*9 (if ORIG A-9*9 (TRANSPOSAR A-9*9)))
  (foreach Y0 '(-1 2 5)
  ........................................

(defun T+D<12 ()
  (foreach LL (list A-9*9 (TRANSPOSAR A-9*9))
    (if (not POST)
  ........................................
```

en el cas d'**ACTUALITZA-PLUS** i la seva auxiliar **NC<2** serà obligat, no pas per les sis substitucions sinó perquè, mentre l'autor bregava amb els 6 supòsits que han acabat donant lloc a les funcions anteriors, va descobrir accidentalment que el dispositiu tapaforats tenia un forat conceptual molt gran sense tapar.

Si recuperem els casos **0-0 ≡ 0-0***, **F-1 ≡ C-1*** i **C-1 ≡ F-1*** (dels quals ens limitem a reproduir **0-0** a l'esquerra i **F-1** al mig) veureu que no cobreixen, ni de lluny, l'eclosió de caselles amb un valor predeterminat per concurrència dels altres vuit (l'altra tipologia de forats, per la presència del mateix valor a files o columnes veïnes, es recollia en els casos precedents): en teniu la prova a la representació esquemàtica de la dreta, on ja no podem parlar de fila, columna o requadre 3×3 als qui només els quedi per omplir una casella, però on la 5,5 només admet el valor **9**.

```
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X 3 X   X X X
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X 4 X   X X X
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X 5 X   X X X

X X X   0 1 2   X X X       X X X   X X X   X X X       X X X   0 X X   X X X
X X X   3 4 5   X X X       0 1 2   3 4 5   6 7 8       0 1 2   X 0 X   X 7 X
X X X   6 7 8   X X X       X X X   X X X   X X X       X X X   6 X 8   X X X

X X X   X X X   X X X       X X X   X X X   X X X       X X X   X X X   X X X
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X 0 X   X X X
X X X   X X X   X X X       X X X   X X X   X X X       X X X   X X X   X X X
```

És clar que sempre podríem tapar el forat de programació amb un pedaç: consistiria en afegir a les 5 exploracions (entre **(ACTUALITZA-PLUS T)** i **(ACTUALITZA-PLUS ())**) una sisena que revisés totes les caselles lliures, comprovant-hi la compatibilitat primària dels valors 1... 9 i procedint a emplenar-les quan només n'hi hagués un, de compatible. Però ben pensat l'acció seria redundant, perquè el nou dispositiu també detectaria els casos **0-0 ≡ 0-0***, **F-1 ≡ C-1*** i **C-1 ≡ F-1***, raó per la qual més que afegir podríem substituir l'exploració per files, columnes i requadres 3×3 pel nou dispositiu, i així tot quedaria reduït a 3 exploracions.

Doncs tampoc, perquè sí que és veritat que amb tres exploracions en tindríem prou, però no cal que cap dispositiu nou se'n faci càrrec, perquè la feina que volíem adjudicar-li ja és feta: ¿us en recordeu, que **LLL** està actualitzada permanentment? Potser ens havíem dedicat massa a destriar la casuística dels forats detectables amb l'exploració dels requadres 9×3 i 3×9, centrant-nos en ****9*9**** (ara **A-9*9**), i se'ns havia passat per alt la informació valuosa que emmagatzema **LLL** (ara **A-LLL**), però només cal que ens adonem que, en els tres exemples de dalt, els forats tenen en comú l'aspecte de l'element d'**A-LLL** corresponent, que és

(nil nil nil nil nil nil nil nil 9)

per veure que tot plegat consistirà a rastrejar aquesta pseudomatriu, descartant-ne els elements **nil**, fins a localitzar alguna llista amb només vuit membres **nil**:

```
(defun ACTUALITZA-PLUS (ORIG / M N CC NL NC 2X 2Y X1 X2 Y1 Y2)
  (if ORIG
      (progn
        (setq Y1 -1)
        (foreach F A-LLL
          (if (not PLUS)
              (progn
                (setq Y1 (1+ Y1) X1 -1)
                (foreach E F
                  (if (not PLUS)
                      (progn
                        (setq X1 (1+ X1))
                        (if E
                            (progn
                              (setq K 0)
                              (foreach M L1A9
                                (if (and (< K 2) (member M E))
                                    (setq N M K (1+ K))))
                              (if (= K 1)
                                  (progn
                                    (setq K (list X1 Y1))
                                    (PLUS-N->P)))))))))))))
```

```
  (foreach X1 '(0 3 6)
    (setq M (cons (SUB-X*Y A-9*9* (list X1 0) (list (+ X1 2) 8)) M)))
  (setq M (reverse M))
  (foreach Y1 '(0 3 6)
    (if (not PLUS)
        (progn
          (setq L (SUB-X*Y A-9*9* (list 0 Y1) (list 8 (+ Y1 2))))
          (foreach N L1A9
            (setq LL (E-MEMBER N L (list 0 Y1)) NL (length LL))
            (if (= NL 1)
                (progn
                  (setq 2X (mapcar '* (3-1 (/ (caar LL) 3)) '(3 3))
                        2Y (mapcar '+ (3-1 (rem (cadar LL) 3)) (list Y1 Y1)))
                  (foreach X 2X
                    (if (and (not PLUS)
                             (setq CC (E-MEMBER N (nth (/ X 3) M) (list X 0))
                                   NC (length CC))
                             (< NC 2)
                             (or ORIG (= NC 0)))
                        (NC<2))))
                (if (= NL 2)
                    (progn
                      (setq X1 (* (3-2 (list (/ (caar LL) 3) (/ (caadr LL) 3))) 3)
                            Y1 (+ Y1 (3-2 (list (rem (cadar LL) 3)
                                                (rem (cadadr LL) 3))))
                            CC (E-MEMBER N (nth (/ X1 3) M) (list X1 0))
                            NC (length CC))
                      (if (< NC 3)
                          (if (= NC 2)
                              (if ORIG
                                  (progn
                                    (setq X1 (+ X1 (3-2 (list (rem (caar CC) 3)
                                                              (rem (caadr CC) 3)))))
                                    (if (listp (setq K (nth X1 (nth Y1 A-9*9))))
                                        (PLUS-N->P))))
                              (NC<2)))))))))))
```

I no ens oblidem de **NC<2**, encara que només sigui per l'anunciada substitució de ****9*9**** per **A-9*9**:

```
(defun NC<2 ()
  (if (= NL 2) (setq X2 (1- X1)))
  (setq X1 (if (= NC 1) (caar CC)) I 0)
  (foreach Y (if (= NL 1) 2Y (list Y1))
    (if (= NL 1) (setq X2 (1- X)))
    (repeat 3 (if (and (/= (setq X2 (1+ X2)) X1)
                       (listp (setq J (nth X2 (nth Y A-9*9)))))
                  (setq K J I (1+ I)))))
  (if (= I 1) (PLUS-N->P)))
```

Etapa prèvia: crear una solució, II (la difícil simplicitat)

Ja que hem hagut de retornar a la casuística de **T+D<12** i en particular al **supòsit 6**, per tal d'analitzar una situació encara més concreta que ens feia grinyolar una versió d'**ACTUALITZA** que havíem donat per bona, ens permetrem seguir ajornant les explicacions pendents sobre aspectes secundaris de **MASCARA** i, tot admetent que el mètode seguit a la primera d'aquestes funcions mai no li ha arribat a fer el pes, l'autor es compromet a anar fins on calgui per assegurar la solidesa de fonaments. Abans ha admès que la formulació dels 6 supòsits (dels 6 supòsits homologats fins ara, per ser exactes) havia estat un inacabable degoteig de casos inesperats quan semblava que ja havia parat de ploure. Amb el sisè (el que hem vist en quart lloc) va començar a sospitar que n'hi podien haver força més i que seguir buscant casos particulars, depenent tant de l'atzar i sense garanties que l'última troballa fos la definitiva, era un mètode poc consistent. Potser calia cercar un procediment més arrelat en els fonaments teòrics del procés i deixar clares algunes qüestions que acceptàvem implícitament però sense ser massa conscients dels seus límits:

- Que hi havia un remei infal·lible però que en la pràctica no semblava assumible: fer **POST** = **T** d'entrada (és a dir, usar **RE-MEMBER** de bon començament). Perquè si ens estem trencant les banyes per fabricar un dispositiu que permeti detectar a temps situacions que durien a un carreró sense sortida és precisament per evitar que **RE-MEMBER** hagi d'actuar en un escaquer pràcticament buit, circumstància que es traduiria en llarguíssims temps d'espera, potser no els primers (que tindrien molts graus de llibertat però pocs condicionaments) però sí amb poques caselles plenes (hauríen baixat els primers però pujarien més ràpidament els segons).
- Com vèiem en la pàgina 142, **RE-MEMBER** explora la situació actual de l'escaquer incloent-hi la casella **P** que anem a emplenar, on hi va provant tots els valors **N** compatibles en primera instància, que tenim a **(nth (car P) (nth (cadr P) LLL))**. Amb cada **N** intentarà emplenar ordenadament les caselles lliures i, si arriba a alguna que no té cap valor compatible, remunta la seqüència fins a trobar algun valor compatible encara no utilitzat i tornem a emplenar. El procés acaba quan:
  A) Arriba a omplir l'última casella buida (**N** és compatible *stricto sensu*).
  B) Arriba a una casella que no té cap valor compatible i tampoc no queden en les precedents valors compatibles que no hagi assajat (**N** no és compatible).
- Si en comptes de perseguir l'acreditació dels valors candidats (**A**), utilitzéssim **RE-MEMBER** per activar **POST** així que algun candidat es revelés inviable (**B**), ¿què passaria? Doncs que, amb **POST** = **T**, hauríem de seguir el pla previst: seguir una exploració exhaustiva perquè afloressin els valors estrictament assignables; en definitiva i presentat retòricament com si fos una cosa diferent, estaríem fent el mateix que plantejàvem críticament en el primer apartat. I és que **RE-MEMBER** ha estat concebut per anar fins a les últimes conseqüències (l'autorecursió n'és l'estructura idònia) i no per fer prospeccions limitades a un nombre de passes preestablert.
- Els sis supòsits individualitzats fins ara ens serveixen per evitar assignacions errònies (amb efectes fatals i irreversibles) en la jugada següent. Si amb això ja n'hi ha prou (acceptem-ho provisionalment, que més endavant contextualitzarem l'afirmació), totes aquestes situacions serien fàcils d'identificar al marge de l'especificitat de cada cas, però no pas a ****9*9**** sinó a **LLL**. N'hi hauria prou a considerar totes les possibilitats que té l'usuari a l'hora d'emplenar una casella més (emplenar cada casella lliure amb tots els valors compatibles en primera instància) i veure com quedaria **LLL** després d'haver dut a terme cada una d'aquestes possibilitats; així que hi descobríssim alguna configuració de mal averany, ja podríem interrompre l'exploració i activar **POST**.
- De fet, ja hi ha un precedent sobre quan fecunda s'ha revelat la intuició precoç d'utilitzar i mantenir actualitzada no només informació sobre el valor assignat a cada casella sinó sobre els valors assignables: ens referim a l'última versió d'**ACTUALITZA-PLUS**, que clarament avantatja en simplicitat i eficàcia l'anterior, basada (com aquesta de **T+D<12** que pretenem superar) en un rastreig de situacions particulars que mai no proporciona la plena convicció interna de ser exhaustiva. No seria possible una revolució similar?
- La qüestió és quins són aquests signes estranys que hauríem de localitzar a **LLL** per saber que el sudoku està condemnat. Sabem que la manifestació visible del col·lapse és no poder emplenar una casella (simultàniament pot passar a diverses caselles) perquè ja no admet cap valor 1... 9 d'assignació, situació que a **LLL**

es manifesta en l'existència d'elements **(nil nil nil nil nil nil nil nil nil)**, però l'error fatal, que tard o d'hora ens hi portarà, es cova abans i es dóna amb situacions com la dels nostres 6 supòsits, on habitualment hem representat en **negreta** l'assignació que ens ho pot engegar tot a dida. Així, en l'exemple de l'esquerra del **supòsit 1**, si arribem a introduir el valor **7** ja haurem begut oli, perque si després d'això volem completar la fila inferior només podrem disposar de dos valors (**8** i **9**) per a la 7ª casella, un (**8** o **9**) per a la 8ª i cap per a la 9ª (que estaria representada a **LLL** per **(nil nil nil nil nil nil nil nil nil)**). Si, per tocar un cas diferent, passem a l'exemple de la dreta del **supòsit 4** i un cop introduït a la casella central el fatídic **3** passem a la casella 4,5, tindrem al nostre abast 8 valors per assignar (tots menys el **3**). Si hi introduïm **1** o **2** i seguim, descrivint un gir de sentit antihorari a l'entorn de 5,5, disposarem de 6 valors a la 4,4 (tots menys **1**, **2** i **3**), de 5 a la 5,4, de 4 a la 6,4, de 3 a la 6,5, de 2 a la 6,6 i d'un a la 5,6, però no en disposarem de cap per a la 4,6; i si, en comptes d'emplenar 5,4 amb **1** o **2**, ho fem amb un valor de **4** a **9**, només en disposarem de 5 a la 4,4, de 4 a la 5,4, de 3 a la 6,4, de 2 a la 6,5, d'un a la 6,6, però no en disposarem de cap per a la 5,6 ni per a la 4,6. Quan això passa en temps real d'assignació de valors definitius, és la fi, però no quan el valor **7** a la casella 7,3 del requadre 9×3 (exemple esquerre del **supòsit 1**) o el valor **3** a la casella 5,5 de l'escaquer 9×9 (exemple dret del **supòsit 6**) són innòqües prospeccions a un pas per endavant: l'únic que cal és identificar a **LLL** alguna configuració perceptible abans que la prospecció hi faci aparèixer els fatídics elements **(nil nil nil nil nil nil nil nil nil)**.

L'exploració de **LLL** cal realitzar-la per files, columnes i requadres 3×3, però per no fer massa feixuga l'exposició parlarem com si la limitéssim als requadres 3×3, exploració que d'altra banda seria suficient en una àmplia majoria de casos (de fet, únicament s'han detectat situacions en què la prospecció per requadres no ha revelat cap anomalia però sí que ha saltat l'alarma en fer-la posteriorment per files o per columnes, si just abans havia actuat **ACTUALITZA-PLUS** omplint forats). En el marc de cada un d'aquests tres conjunts hi ha 3 configuracions típicament "precatastròfiques", tot i que, si no es dóna la 1ª, la 2ª i la 3ª es presenten sempre juntes perquè, com el *yin* i el *yang*, són manifestacions complementàries d'una mateixa realitat:

1) Tots els valors 1... 9 hi han de sortir almenys una vegada, directament com a elements de **A-9*9** o com a part dels elements no nuls de **A-LLL** (passant per alt l'estructura de pseudo-matriu 3×3 i pensant en termes d'autèntica matriu, com si haguéssim escollit jugar amb les llistes-vectors files/columnes). En altres paraules, els 9 enters hi han de figurar, com valors assignats a unes caselles o com valors assignables, en primera instància, a les que encara resten buides. Si considerem l'exemple central del **supòsit 1**, tot suposant que és el requadre 9×3 inferior i que hem introduït el valor **7** a la casella 6,2, els elements de **A-9*9** (esquerra) i els de **A-LLL** (dreta, amb el valor **nil** representat per zeros, per estalviar espai) corresponents al requadre 3×3 del mig seran:

(3 2)	(4 2)	(5 2)	(0 0 0 0 0 0 0 **8 9**)	(0 0 0 0 0 0 0 **8 9**)	(0 0 0 0 0 0 0 **8 9**)
4	**5**	**6**	0	0	0
1	**2**	**3**	0	0	0

   Observeu que els enters **1**, **2**, **3**, **4**, **5** i **6** apareixen directament com a elements de **A-9*9** (recordeu que els elements (3 2) (4 2) i (5 2) corresponen a caselles buides i en representen les coordenades), mentre que **8** i **9** surten per triplicat a **A-LLL**, dins de les llistes de valors compatibles. El **7** no surt enlloc, perquè la introducció d'aquest mateix valor a la casella veïna 6,2 ho esguerra tot.

2) Amb tots els subconjunts binaris, ternaris,... **N**-aris (**1 < N ≤ 9**) de llistes de **A-LLL** que es puguin formar, quan apleguen **M** valors 1... 9 (**1 ≤ M ≤ 9**) diferents s'ha de complir **M ≥ N**: si amb un subconjunt de **N** llistes aplegant **M** valors no nuls i diferents s'esdevingués que **M** < **N**, això implicaria que algunes de les caselles corresponents a aquestes llistes es quedaria sense valor d'assignació. Això ja passava a l'exemple precedent (línia superior de **A-LLL**, amb 3 llistes (0 0 0 0 0 0 0 **8 9**) que només contenen dos enters diferents i no nuls), a més d'incomplir-se la **norma 1**, però tornant al **supòsit 1**, ara per anar a l'exemple de l'esquerra, veureu com a la fila inferior de **A-LLL** (les mateixes 3 llistes (0 0 0 0 0 0 0 **8 9**)) és **M** < **N**, tot i no infringir-se la **norma 1** (a l'esquerra,

en **A-9*9**, hi surt el valor **7** que faltava a la dreta). El lector ja haurà entès que el requadre 3×3 representat ara és l'inferior-dret:

```
  7    (7 2) (8 2) |          0          (1 2 3 4 5 6 0 8 9) (1 2 3 4 5 6 0 8 9)
                   |
(6 1) (7 1) (8 1) | (1 2 3 4 5 6 0 8 9) (1 2 3 4 5 6 0 8 9) (1 2 3 4 5 6 0 8 9)
                   |
(6 0) (7 0) (8 0) | (0 0 0 0 0 0 0 8 9) (0 0 0 0 0 0 0 8 9) (0 0 0 0 0 0 0 8 9)
```

3) Amb tots els subconjunts binaris, ternaris,... **N**-aris (**1 < N ≤ 9**) que es puguin formar utilitzant valors 1... 9 presents a les llistes de **A-LLL**, quan els seus elements apareixen a **M** d'aquestes llistes (**1 ≤ M ≤ 9**), s'ha de complir **M ≥ N**: si amb un subconjunt de **N** valors presents a **M** llistes s'esdevingués **M** < **N**, això implicaria que algun d'aquests valors es quedaria sense assignar a cap casella. Això ja passava en l'exemple precedent, a més d'incomplir-se la **norma 2**: els 6 valors **1**, **2**, **3**, **4**, **5** i **6** només surten a 5 llistes (les que quedaven). I és que, quan es compleix la **norma 1** però no la **2** [**3**], tampoc no es compleix la **3** [**2**], perquè són faves comptades: en qualsevol fila, columna o requadre 3×3 el nombre de valors 1... 9 encara no assignats ha de coincidir amb el de caselles buides, i els valors assignables a cada casella poden oscil·lar entre un (això si fem prospeccions avançades, perquè en temps real **ACTUALITZA-PLUS** ja hauria emplenat aquestes caselles) i el total de valors pendents d'assignació; ara bé, mai no podrà ser que més d'una casella només tingui un valor assignable i que aquest valor sigui el mateix, que més de dues només tinguin un dos valors assignables i que aquests valors siguin els mateixos, que més de tres només tinguin tres valors assignables i que aquests valors siguin els mateixos (totes aquests dos o fins i tot alguna només un),... o que en el conjunt de les caselles buides hi apareguin tots els valors no assignats menys un, perquè en tots aquests casos queda una casella buida i un valor sense assignar.

Potser amb un exemple senzill tot quedarà més clar. Imagineu un requadre 3×3 amb 5 caselles emplenades (amb els valors **1**, **2**, **3**, **4** i **5**) i 4 de buides, que anomenarem **A**, **B**, **C**,i **D**. Totes aquestes possibilitats són acceptables:

- **A**) valors assignables: **6**, **7**, **8**, **9**
  B) valors assignables: **6**, **7**, **8**, **9**
  C) valors assignables: **6**, **7**, **8**, **9**
  D) valors assignables: **6**, **7**, **8**, **9**

- **A**) valors assignables: **6**, **7**, **8**, **9**
  B) valors assignables: **6**, **7**, **8**
  C) valors assignables: **7**, **8**, **9**
  D) valors assignables: **7**, **8**, **9**

- **A**) valors assignables: **6**, **7**, **8**
  B) valors assignables: **6**, **7**, **8**
  C) valors assignables: **7**, **8**, **9**
  D) valors assignables: **7**, **8**, **9**

- **A**) valors assignables: **6**, **7**, **8**, **9**
  B) valors assignables: **6**, **7**, **8**
  C) valors assignables: **6**, **7**
  D) valors assignables: **8**, **9**

- **A**) valors assignables: **6**, **7**
  B) valors assignables: **6**, **7**
  C) valors assignables: **8**, **9**
  D) valors assignables: **8**, **9**

Les que venen ara no ens han de preocupar, tot i semblar contradictòries, perquè si l'assignació prospectiva es confirmés, **ACTUALITZA-PLUS** abordaria automàticament l'emplenament de forats (caselles amb el valor ja determinat):

- **A**) valors assignables: **6**, **7**, **8**, **9**
  B) valors assignables: **7**, **8**, **9**
  C) valors assignables: **8**, **9**
  D) valors assignables: **9**

- **A**) valors assignables: **6, 8, 9**
  B) valors assignables: **7, 8, 9**
  C) valors assignables: **8, 9**
  D) valors assignables: **8, 9**

- **A**) valors assignables: **6**
  B) valors assignables: **7**
  C) valors assignables: **8, 9**
  D) valors assignables: **8, 9**

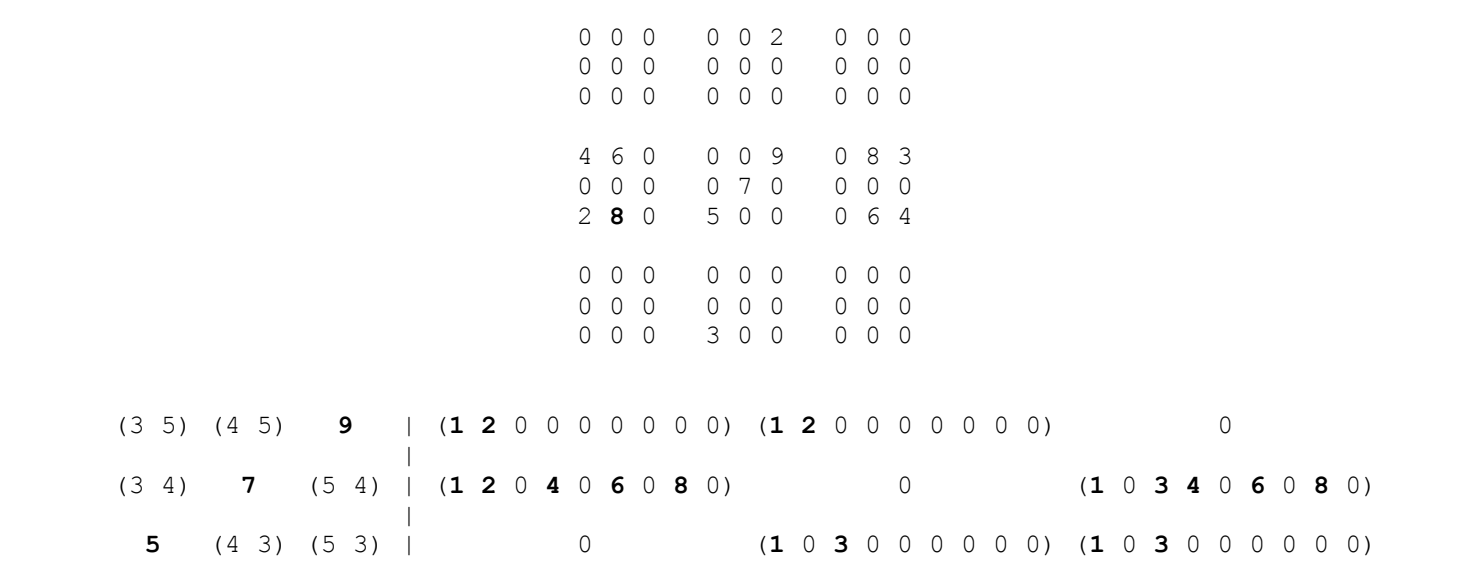
Per això ens hem permés en la **norma 3** ignorar els subconjunts d'un sol element no nul. I pel que fa a subconjunts d'una sola llista amb tots el elements nuls, en la **norma 2**, ja hem dit que quan ens trobem amb un element (0 0 0 0 0 0 0 0 0) (o bé **(nil nil nil nil nil nil nil nil nil)**, en la notació original) ja serà massa tard. El que cal és detectar configuracions simptomàtiques prèvies, com:

- **A**) valors assignables: **6**
  B) valors assignables: **6**
  C) valors assignables: **7, 8, 9**
  D) valors assignables: **7, 8, 9**

- **A**) valors assignables: **6, 7**
  B) valors assignables: **8, 9**
  C) valors assignables: **8, 9**
  D) valors assignables: **8, 9**

Del primer tipus l'autor no ha tingut la sort de trobar-ne exemples però tampoc no troba cap raó per descartar-ne l'existència, i com que verificar si en algun dels subconjunts de llistes (que haurem hagut de formar de tota manera) aquestes només tenen un element no nul té una incidència mínima sobre la durada de l'exploració, ha decidit de mantenir-ho. Si configuracions com aquesta arriben a produir-se, sí que podem assegurar que l'aparició de dues llistes (0 0 0 0 0 **6** 0 0 0) ha de ser simultànea (és a dir, no existir cap de les dues abans d'un emplenament, i sorgir de cop les dues després), perquè si primer n'aparegués només una, **ACTUALIZA-PLUS** (el seu primer dispositiu) ompliria el forat i ja no podria aparèixer la segona.

Del segon tipus en són els dos exemples utilitzats per il·lustrar les tres normes, i si recuperéssim la resta dels que pàgines enrera ens van servir en la descripció dels 6 supòsits de la versió superada de **T+D<16**, veuríem que responen a la mateixa tipologia. Però, com que sempre anem a parar a un conjunt de llistes iguals (**B**, **C** i **D**), amb la resta de valors assignables aplegats a l'única llista que en queda al marge (**A**), afegirem un exemple per deixar clar que no forçosament ha de ser així. Si considerem la situació representada, on el **8** en negreta representa la casellla en què es fa la prospecció i que aquest valor és un dels que ha d'activar l'alarma **POST** = **T**, sota l'escaquer 9×9 veureu els fragments de **A-9*9** (esquerra) i de **A-LLL** (dreta, amb els **nil** representats per zeros, com abans) corresponents al requadre 3×3 central:



D'una banda veureu 4 llistes (les 2 superiors i les 2 inferiors) que en conjunt només apleguen 3 valors diferents i no nuls (**1**, **2** i **3**), i de l'altra 3 valors (**4**, **6** i **8**) només presents en dues llistes (centre). S'acompleix la **norma 1** perquè els valors absents de **A-LLL**, **5**, **7** i **9**, són a **A-9*9**, pero entre aquests hi ha ha els 3 trànsfugues **4**, **6** i **8** que s'han passat del subconjunt de 4 llistes (podem imaginar que inicialment dues eren (**1** **2** 0 **4** 0 **6** 0 0 0) i dues eren (**1** 0 **3** 0 0 **6** 0 **8** 0)) al de 2 (podem imaginar que eren (**1** **2** 0 0 0 0 0 0 0 0) i (**1** 0 **3** 0 0 0 0 0 0)), trencant simultàniament les **normes 2** i **3**. En el segon dels exemples més senzills d'abans també quedava palès l'origen de la descompensació i com transferint algun valor d'un a un altre subconjunt de llistes (allà, passant tots els valors **1**, **2**, **3**, **4**, **5** o **6** a alguna de les 3 llistes (0 0 0 0 0 0 0 **8** **9**)) tot es reequilibrava.

Allò que no podem perdre de vista és que la complementarietat de les **normes 2** i **3** (o equivalència lògica, si ho preferiu així) només es dóna sota la hipòtesi que la **norma 1** s'acompleix; si no és així, ni tan sols té sentit referir-se a la **norma 3**:

- **Norma 1** -> { **norma 2** <--> **norma 3** ; no **norma 2** <--> no **norma 3** }

- No **norma 1** -> no **norma 2** (**norma 2** -> **norma 1** -> **norma 3**)

Si ignorem la **norma 1** i ampliem la **norma 3** considerant en el requadre 3×3 de **A-LLL** tots els valors 1... 9 no coberts per **A-9*9** (incloent-hi els absents per infracció de la **norma 1**) i no només, com fèiem, els que apareixen en les llistes de **A-LLL**, llavors podrem generalitzar a relació dual la que lliga la **norma 2** amb la **norma 3**: n'hi hauria prou a incloure en la seva definició els subconjunts d'un únic element (**0 < N ≤ 9**) i comptabilitzar com **0** la inexistència de llistes amb aquest contingut (**0 ≤ M ≤ 9**), de forma que en l'exemple utilitzat per il·lustrar la **norma 1** diríem que no solament incompleix la **norma 2** (3 llistes (0 0 0 0 0 0 0 **8** **9**) amb 2 valors diferents) sinó la **norma 3** (1 valor en 0 llistes (0 0 0 0 0 0 **7** 0 0)). Entenent-ho d'aquesta manera, l'equivalència d'ambdues normes es pot esquematitzar així:

- N**orma 2** <--> **norma 3**

- No **norma 2** <--> no **norma 3**

Però no surt a compte complicar-se tant la vida, perquè si ens mantenim en les definicions inicials tot és més senzill: si no s'acompleix la **norma 1**, ja som al cap del carrer, i si s'acompleix n'hi haurà prou a veure si s'acompleix la **norma 2** o la **norma 3**. També és cert que ens podríem limitar a verificar la **norma 2** (si no s'acompleix ja som al cap del carrer, i si s'acompleix s'acompleix la **norma 1** i, amb aquestes dues, s'acompleix també la **norma 3**) però, valoracions estètiques a banda, resulta més pràctic despatxar casos com el que il·lustrava la **norma 1** amb un algorisme específic però simple que no pas dispensar un tractament genèric que trigarà més a detectar-lo. I, com que ja hem especulat prou, anem a posar tot això en solfa.

```
(defun EXPL-SUBCONJ (NUM / LA I NO)
  (if NUM
     (foreach N (reverse L1A9)
       (setq NO T)
       (foreach E L
         (if (and NO (member N E))
             (setq NO () LA (cons (list N) LA)))))
     (repeat (setq N (length L))
       (setq N (1- N) LA (cons (list N) LA))))
  (setq LL LA N 1)
  (repeat (1- (length LL))
    (if (not POST)
        (progn
          (setq N (1+ N) LA LL LL ())
          (repeat (1- (length LA))
            (setq F (car LA) LA (cdr LA) I (reverse (cdr (reverse F))))
            (foreach E LA
              (if (equal (reverse (cdr (reverse E))) I)
                  (setq LL (cons (append F (list (last E))) LL)))))
```

```
          (setq LL (reverse LL))
          (foreach F LL
            (if (not POST)
                (progn
                  (if NUM
                      (progn
                        (setq M 0)
                        (foreach E L
                          (setq NO T)
                          (foreach I F
                            (if (and NO (member I E)) (setq M (1+ M) NO ()))))))
                      (progn
                        (setq LA ())
                        (foreach I F
                          (foreach E (nth I L)
                            (if (not (member E LA))
                                (setq LA (cons E LA)))))
                        (setq M (length LA))))
                  (if (< M N) (setq POST T)))))))))

(defun EXPLORA-3*9 (/ X Y 9+1 L+1 L LL M N)
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not POST)
          (progn
            (setq 9+1 (SUB-X*Y A-9*9+1 (list X Y) (list (+ X 2) (+ Y 2)))
                  L+1 (SUB-X*Y A-LLL+1 (list X Y) (list (+ X 2) (+ Y 2))) L ())
;           *********************************************************************
            (foreach F 9+1 (foreach E F (if (atom E) (setq L (cons E L)))))     ;*
            (foreach F L+1                                                      ;*
              (foreach E F                                                      ;*
                (if E (foreach N E                                              ;*
                        (if (and N (not (member N L))) (setq L (cons N L)))))) ;*
            (if (< (length L) 9) (setq POST T))                                 ;*
;           *********************************************************************
            (if (not POST)
                (progn
                  (setq L ())
                  (foreach F L+1
                    (foreach E F
                      (if E (progn
                              (setq M ())
                              (foreach N E (if N (setq M (cons N M))))
                              (setq L (cons M L))))))
                  (EXPL-SUBCONJ ()))))))))
  (setq X ())
  (repeat 2
    (if (not POST)
        (progn
          (if X (setq A-9*9+1 (TRANSPOSAR A-9*9+1)
                      A-LLL+1 (TRANSPOSAR A-LLL+1)))
          (setq X T Y -1)
          (repeat 9
            (if (not POST)
                (progn
                  (setq Y (1+ Y) 9+1 (nth Y A-9*9+1) L+1 (nth Y A-LLL+1) L ())
;                 *************************************************************
                  (foreach E 9+1 (if (atom E) (setq L (cons E L))))             ;*
                  (foreach E L+1                                                ;*
                    (if E (foreach N E                                          ;*
                            (if (and N (not (member N L)))                      ;*
                                (setq L (cons N L)))))                          ;*
                  (if (< (length L) 9)                                          ;*
                      (setq POST T)                                             ;*
                      (progn                                                    ;*
                        (setq L ())                                             ;*
;                 *************************************************************
```

```
                        (foreach E L+1
                          (if E (progn
                                  (setq M ())
                                  (foreach N E (if N (setq M (cons N M))))
                                  (setq L (cons M L)))))
                        (EXPL-SUBCONJ ()))))))))))))

(defun EXPLORA+1 (/ X Y A-N E F A-9*9+1 A-LLL+1)
  (foreach K A-9*9
    (if (not POST)
        (foreach J K
          (if (and (not POST) (listp J))
              (progn
                (setq X (car J) Y (cadr J))
                (foreach A-N (ELEMENT A-LLL)
                  (if (and (not POST) A-N)
                      (progn
                        (setq A-9*9+1 () E -1)
                        (foreach I A-9*9
                          (setq E (1+ E)
                                F (if (= E Y) (subst A-N J I) I)
                                A-9*9+1 (cons F A-9*9+1)))
                        (setq A-9*9+1 (reverse A-9*9+1)
                              A-LLL+1 (ACT-LLL X Y A-LLL))
                        (EXPLORA-3*9)))))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS NOU E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST)
             (progn
               (EXPLORA+1)
               (setq NOU POST)))
         (if REAL
             (progn
               (ACTUALITZA-PLUS T)
               (if (not PLUS)
                   (progn
                     (setq A-9*9 (TRANSPOSAR A-9*9))
                     (ACTUALITZA-PLUS ())
                     (setq A-9*9 (TRANSPOSAR A-9*9))))
               (if (and PLUS NOU) (setq POST () NOU ()))))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))
```

Comentant els mòduls funcionals cap enrera, l'única novetat d'**ACTUALITZA** és la substitució de **(T+D<12)** per **(EXPLORA+1)**: com que la nova funció prescindeix dels 6 supòsits particularistes, del primer dels quals (que en tots els requadres 9×3 o 3×9 la suma del nombre de caselles plenes i de valors diferents havia de romandre inferior a 12) en manllevava el nom, adoptarem un nom nou que suggereixi que anem explorant l'escaquer 9×9 una casella per endavant de l'última realment emplenada. Pel que fa a **EXPLORA+1**, la seva missió serà anar obtenint les pseudomatrius 9×9 **A-9*9+1** y **A-LLL+1** (actualitzacions de **A-9*9** y **A-LLL**, respectivament) corresponents a diferents hipòtesis d'emplenament (de cadascuna de les caselles lliures amb tots els valors 1... 9 compatibles en primera instancia), fins que la detecció eventual d'alguna de les configuracions anòmales que hem condensat en l'incompliment de les **normes 1**, **2** o **3** (funció **EXPL-SUBCONJ**, que mitjançant **EXPLORA-3*9**, és aplicat a nou requadres 3×3, nou files i nou columnes) activi la variable **POST**.

Abans de seguir, però, advertirem que **EXPL-SUBCONJ** té l'argument binari **NUM** que permet utilitzar-lo per comprovar la **norma 3** (si és **T**, i aleshores **EXPL-SUBCONJ** formarà totes les combinacions **N**-àries amb els valors 1... 9 representats en les llistes de **L+1** i localitzarà les **M** llistes en què hi hagi algun d'aquests valors) o la **norma 2** (si és **nil**, i llavors formarà totes les combinacions **N**-àries amb les

llistes de **L+1** i localitzarà els **M** valors 1... 9 diferents que hi hagi en aquestes llistes), després d'haver comprovat l'acompliment de la **norma 1**. També recordarem que la verificació prèvia de la **norma 1** és obligada si fem **(EXPL-SUBCONJ T)** però en podem prescindir (esborrant el codi comprès entre les dues línies d'asteriscs) si fem **(EXPL-SUBCONJ ())**, tot i ser aconsellable mantenir-la per raons pràctiques.

Aclarit això, sigueu sincers: ¿us fan patxoca les dues crosses que té **ACTUALITZA**, **EXPLORA+1** i **ACTUALITZA-PLUS**, cadascuna amb les seves funcions derivades? Perquè la veritat és que semblen models d'èpoques diferents: comparant **T+D<12** amb la segona encara podíem dir que les dues carraques s'inspiraven en una mateixa metodologia casuística que, per aquest mateix caràcter, ens deixava amb la sensació permanent d'haver quedat amb el cul a l'aire. Ara, amb el pas de **T+D<12** a **EXPLORA+1**, deixant la constatació immediata i assolint un major grau d'abstracció, tenim la seguretat d'estar sota cobert, però amb el dispositiu tapaforats no hem donat el mateix salt qualitatiu... o ens hem quedat a mitges. A mitges? ¿Que potser **ACTUALITZA-PLUS** no pecava d'un particularisme tant o més exacerbat que el de **T+D<12**? Sí, però quan ja erem a acabar el subcapítol precedent recordareu que ens va venir la inspiració i vam adonar-nos que els 11 casos catalogats no cobrien una situació tan arquetípica de casella amb el valor ja determinat com l'existència en **A-LLL** d'algun element de l'estil de **(nil nil nil nil 5 nil nil nil nil)**, en què sols un membre de la llista fos no nul. Doncs bé, ara podríem reformular una de les reflexions amb què obríem el present subcapítol, però no en relació als 6 supòsits que acabaríem substituint per 3 normes sinó a les 11 situacions que acusen la presència de forats per tapar: ja que vam tenir l'ocurrència de treballar amb pseudomatrius 9×9 actualitzades per disposar dels valors assignables a les caselles (**LLL**, **A-LLL**, etc.), treure'ls suc.

Allò de voler és poder, màxima benintencionada que sovint serveix més per frustrar que per altra cosa, aquí funciona prou bé, perquè n'hi ha prou a preguntar-se com una casella que només admetés un valor **A-N** hauria de quedar representada en **A-LLL** per trobar la resposta: si no és de les detectades pel filtre inicial que fa poc esmentàvem (una llista amb el valor **A-N** i els altres 8 membres **nil**), es distingirà per ser l'única llista del seu requadre 3×3 on aparegui aquest valor (suposant que **A-N** = **5**, una llista tipus **(nil nil 3 nil 5 nil nil 8 nil)**); els altres 8 elements seran **nil** o llistes on no hi figura **A-N**. Cal insistir en què la implementació d'un dispositiu que faci aquesta cerca per requadres 3×3 no ha d'excloure la prèvia que que ja veníem fent. Imagineu, si no, un mateix requadre on coexisteixi l'element **(nil nil nil nil 5 nil nil nil nil)** amb l'element **(nil nil 3 nil 5 nil nil 8 nil)**, encara que no hi hagi cap altra llista amb el valor **5**: si únicament deixéssim el segon dispositiu, **ACTUALITZA-PLUS** no detectaria res pel que fa al **5**, perquè aquest valor està repetit en el requadre; però si abans apliquem el primer dispositiu, la primera llista serà detectada, **PLUS** s'activarà (i la cerca dins d'**ACTUALITZA-PLUS** s'interromprà) i **PLUS-N->P** s'executa, emplenant amb el **5** la casella corresponent; i a la següent iteració **(while (not PLUS) ...)** d'**ACTUALITZA**, el **5** ja desapareixerà d'aquest requadre 3×3 de **A-LLL**. Aclarit això, veiem-ne la traducció a codi:

```
(defun ACTUALITZA-PLUS (ORIG / 3*3 LL3 M N X1 Y1)
  (if ORIG
      (progn
        (setq Y1 -1)
        (foreach F A-LLL
          (if (not PLUS)
              (progn
                (setq Y1 (1+ Y1) X1 -1)
                (foreach E F
                  (if (not PLUS)
                      (progn
                        (setq X1 (1+ X1))
                        (if E
                            (progn
                              (setq K 0)
                              (foreach M L1A9
                                (if (and (< K 2) (member M E))
                                    (setq N M K (1+ K))))
                              (if (= K 1)
                                  (progn
                                    (setq K (list X1 Y1))
                                    (PLUS-N->P)))))))))))))
```

```
  (if (and ORIG (not PLUS))
      (foreach Y1 '(0 3 6)
        (foreach X1 '(0 3 6)
          (if (not PLUS)
              (progn
                (setq 3*3 (SUB-X*Y A-9*9 (list X1 Y1) (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL (list X1 Y1) (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0 Y2 (1- Y1))
                        (foreach F LL3
                          (setq X2 (1- X1) Y2 (1+ Y2))
                          (foreach E F
                            (setq X2 (1+ X2))
                            (if (member N E) (setq K (1+ K)))))
                        (if (= K 1)
                            (progn
                              (setq K (nth X2 (nth Y2 3*3))))
                              (PLUS-N->P))))))))))))
```

Podem simplificar tots dos dispositius, prescindint dels contadors per identificar els elements homòlegs de **A-9*9** i **A-LLL**, amb una formulació més compacta on l'ús de **(mapcar '(lambda ...) ...)** desplaci el de **(foreach ... (foreach ...))**:

```
(defun ACTUALITZA-PLUS (ORIG / 3*3 LL3 M N X1 Y1)
  (if ORIG
      (mapcar '(lambda (F-9*9 F-LLL)
                 (if (not PLUS)
                     (mapcar '(lambda (E-9*9 E-LLL)
                                (if (and (not PLUS) E-LLL)
                                    (progn
                                      (setq K 0)
                                      (foreach M L1A9
                                        (if (and (< K 2) (member M E-LLL))
                                            (setq N M K (1+ K))))
                                      (if (= K 1)
                                          (progn
                                            (setq K E-9*9)
                                            (PLUS-N->P))))))
                             F-9*9 F-LLL)))
              A-9*9 A-LLL))
  (if (and ORIG (not PLUS))
      (foreach Y1 '(0 3 6)
        (foreach X1 '(0 3 6)
          (if (not PLUS)
              (progn
                (setq 3*3 (SUB-X*Y A-9*9 (list X1 Y1) (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL (list X1 Y1) (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0)
                        (mapcar '(lambda (F-3*3 F-LL3)
                                   (mapcar '(lambda (E-3*3 E-LL3)
                                              (if (member N E-LL3)
                                                  (setq K (1+ K) M E-3*3)))
                                           F-3*3 F-LL3))
                                3*3 LL3)
                        (if (= K 1)
                            (progn
                              (setq K M)
                              (PLUS-N->P)))))))))))
```

Però fixem-nos que el condicionament a **ORIG** és un vestigi de la versió casuística (algunes deteccions era més còmode fer-les sobre la transposada de **A-9*9**), i que ara en podem prescindir, simplificant alhora **ACTUALITZA**:

```
(defun ACTUALITZA-PLUS (/ 3*3 LL3 M N X1 Y1)
  (mapcar '(lambda (F-9*9 F-LLL)
             (if (not PLUS)
                 (mapcar '(lambda (E-9*9 E-LLL)
                            (if (and (not PLUS) E-LLL)
                                (progn
                                  (setq K 0)
                                  (foreach M L1A9
                                    (if (and (< K 2) (member M E-LLL))
                                        (setq N M K (1+ K))))
                                  (if (= K 1)
                                      (progn
                                        (setq K E-9*9)
                                        (PLUS-N->P))))))
                          F-9*9 F-LLL)))
           A-9*9 A-LLL)
  (if (not PLUS)
      (foreach Y1 '(0 3 6)
        (foreach X1 '(0 3 6)
          (if (not PLUS)
              (progn
                (setq 3*3 (SUB-X*Y A-9*9 (list X1 Y1) (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL (list X1 Y1) (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0)
                        (mapcar '(lambda (F-3*3 F-LL3)
                                   (mapcar '(lambda (E-3*3 E-LL3)
                                              (if (member N E-LL3)
                                                  (setq K (1+ K) M E-3*3)))
                                           F-3*3 F-LL3))
                                3*3 LL3)
                        (if (= K 1)
                            (progn
                              (setq K M)
                              (PLUS-N->P)))))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS NOU E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST)
             (progn
               (EXPLORA+1)
               (setq NOU POST)))
         (if REAL
             (progn
               (ACTUALITZA-PLUS)
               (if (and PLUS NOU) (setq POST () NOU ()))))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))
```

És altament improbable que la curiositat del lector l'hagi dut a implementar les últimes funcions proposades per comprovar-ne l'eficiència (de fet, a hores d'ara ja és altament improbable que en quedi algun, de lector), però l'autor dóna fe que la joia d'haver trobat un procediment que cobreix al 100% el risc de ficar-se de peus a la galleda i trobar-se amb escac i mat en el moviment següent (amb **T+D<12**, la por al descobriment fortuït de situacions no previstes era permanent) ha quedat deslluïda perquè, pel que fa a velocitat i malgrat haver simplificat el dispositiu tapaforats **ACTUALITZA-PLUS** < **ACTUALITZA**, la victòria és pírrica: a diferència de **T+D<12**, no és que es noti que el nou missatge *A l'esquerra, CLIC sobre la casella a omplir:* triga un pèl a sortir a l'àrea de text, sinó que tant l'antic com el seu successor *CLIC sobre el nombre que va a la casella marcada:* segueixen presents, i

sembla que el procés s'hagi col·lapsat. Però, ben pensat, això ja s'hagués pogut preveure, perquè partim de **81 × 9 = 729** comprovacions de **A-LLL**, en cada una de les quals tota casella és considerada **3** vegades (com a part d'un requadre 3×3, d'una fila i d'una columna): així doncs, analitzem **81 × 3 = 243** caselles i un total de **729 × 243 = 177.147** valors 1... 9, que es diu aviat. L'autor us confessa que, per realitzar proves comparatives entre aquest mètode i altres que veurem, de vegades ha tirat pel dret i ha fet trampa, conscient que la majoria d'activacions de **POST** es detecten a nivell de requadres 3×3 (en la pàgina 150 ja hem aventurat aquesta estimació), i ha esborrat la segona meitat d'**EXPLORA-3*9** (des de **(setq X ())** fins al final, tancant la primera amb un parèntesi addicional), tot sabent que se la jugava. Però un recurs provisional mai no es pot fer passar com a producte acabat, i encara que aquesta reducció fos fiable al 100% (val a dir que en el subcapítol següent, *Etapa prèvia: crear una solució, III*, ens deixarem de romanços i acabarem limitant-nos als requadres 3×3, com ja ho fem, per exemple, en **ACTUALITZA-PLUS**), amb haver reduït la durada a una tercera part tampoc no hauríem fet gran cosa.

No hi hauria manera de rebaixar el nombre de comprovacions de valors/casella? Sí que n'hi ha una, i prou dràstica: passar de **177.147** a **2.187**, a base de considerar **9** situacions diferents en comptes de **729**. Però reduir a la **81**-èsima part aquesta quantitat no ho farem per art de màgia sinó mitjançant un canvi d'estratègia i de localització d'**EXPLORA-3*9** en el programa: deixant-la a **ACTUALITZA** < **OMPLE-SUDOKU** (ho expressem així per diferenciar aquest accés d'**ACTUALITZA** < **RE-MEMBER** < **MASCARA** < **OMPLE-SUDOKU**, que es produeix quan **POST** = **T**), després d'emplenar una casella i abans d'assenyalar quina és la que volem omplir a continuació, no hi ha més remei que assegurar-nos la jugada explorant totes les caselles lliures, però si esperem a haver marcat la nova casella ens podrem limitat a aquesta. No és sinó traslladar el dispositiu a **MASCARA**, fent-li un lloc al costat de **RE-MEMBER**, amb qui haurà de compartir protagonisme: mentre sigui **POST** = **nil**, serà **EXPLORA-3*9** qui supervisarà els valors 1... 9 que **(member N L)** ha decidit provisionalment que són assignables a la casella actual, però quan **POST** passi a **T** serà **RE-MEMBER** qui se'n farà càrrec.

Pel que fa als temps d'aparició de valors compatibles en el caseller 3×3, quan ho decideixi **EXPLORA-3*9** es notaran, però seran perfectament assumibles i sobretot regulars, tendint a escurçar-se a mesura que l'escaquer es vagi omplint; quan ho faci **RE-MEMBER** seran més curts, en general, però amb un comportament força més imprevisible. Caldrà fer una sèrie d'ajustos: **EXPLORA+1** desapareix d'**ACTUALITZA**, que de moment perdrà tota referència a **POST**, i part del seu contingut ha de passar a **EXPLORA-3*9**; abans d'anar a una depuració del codi, dotarem a aquesta funció de 4 arguments (les pseudomatrius ****9*9**** i **LLL**, l'enter **N** examinat i la casella que anem a emplenar) i, com que ara en l'activació de **POST** cal diferenciar dos moments (d'una banda la detecció d'una configuració anómala per part d'**EXPLORA-3*9** en una casella, amb un **N** concret, i de l'altra el que això comportarà a l'hora de seguir emplenant caselles), cadascun amb una missió específica (desqualificar aquest **N** com a candidat a la casella actual, que seguirem explorant amb **EXPLORA-3*9** però que en passar a les següents substituirem per **RE-MEMBER**), introduirem a **MASCARA** la variable **REPOST**, que de portes enfora prendrà el relleu de **POST** mentre aquesta fa la seva feina a **EXPLORA-3*9**; l'avís *Espereu el missatge CLIC sobre el nombre...*, que emmarca el caseller 3×3 on van apareixent els valors assignables, no únicament entretindrà l'espera de **RE-MEMBER** sinó també de **EXPLORA-3*9**; i **EXPL-SUBCONJ**, per acabar, queda igual (raó per la qual no la treurem aquí de nou). Veiem doncs les funciona afectades (no incloem **OMPLE-SUDOKU**, l'única afectació de la qual és que **NOU** passa a ser-ne variable local):

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if REAL
             (progn
               (ACTUALITZA-PLUS)
               (if (and PLUS NOU) (setq POST ())))
         (setq PLUS (not PLUS)))
  (setq NOU ())
  (list A-9*9 A-LLL))
```

```
(defun EXPLORA-3*9 (A-9*9 A-LLL A-N J / X Y A-9*9+1 A-LLL+1 E F 9+1 L+1 L LL M N)
  (setq POST () A-9*9+1 () E -1)
  (foreach I A-9*9
    (setq E (1+ E) F (if (= E (cadr J)) (subst A-N J I) I)
          A-9*9+1 (cons F A-9*9+1)))
  (setq A-9*9+1 (reverse A-9*9+1)
        A-LLL+1 (ACT-LLL (car J) (cadr J) A-LLL))
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not POST)
          (progn
            (setq 9+1 (SUB-X*Y A-9*9+1 (list X Y) (list (+ X 2) (+ Y 2)))
                  L+1 (SUB-X*Y A-LLL+1 (list X Y) (list (+ X 2) (+ Y 2))) L ())
;           ********************************************************************
            (foreach F 9+1 (foreach E F (if (atom E) (setq L (cons E L)))))     ;*
            (foreach F L+1                                                      ;*
              (foreach E F                                                      ;*
                (if E (foreach N E                                              ;*
                        (if (and N (not (member N L))) (setq L (cons N L))))))) ;*
            (if (< (length L) 9) (setq POST T))                                 ;*
;           ********************************************************************
            (if (not POST)
                (progn
                  (setq L ())
                  (foreach F L+1
                    (foreach E F
                      (if E (progn
                              (setq M ())
                              (foreach N E (if N (setq M (cons N M))))
                              (setq L (cons M L))))))
                  (EXPL-SUBCONJ ()))))))))
  (setq X ())
  (repeat 2
    (if (not POST)
        (progn
          (if X (setq A-9*9+1 (TRANSPOSAR A-9*9+1) A-LLL+1 (TRANSPOSAR A-LLL+1)))
          (setq X T Y -1)
          (repeat 9
            (if (not POST)
                (progn
                  (setq Y (1+ Y) 9+1 (nth Y A-9*9+1)
                                L+1 (nth Y A-LLL+1) L ())
;                 ************************************************************
                  (foreach E 9+1 (if (atom E) (setq L (cons E L))))             ;*
                  (foreach E L+1                                                ;*
                    (if E (foreach N E                                          ;*
                            (if (and N (not (member N L)))                      ;*
                                (setq L (cons N L))))))                         ;*
                  (if (< (length L) 9)                                          ;*
                      (setq POST T)                                             ;*
                      (progn                                                    ;*
                        (setq L ())                                             ;*
;                 ************************************************************
                        (foreach E L+1
                          (if E (progn
                                  (setq M ())
                                  (foreach N E (if N (setq M (cons N M))))
                                  (setq L (cons M L)))))
                        (EXPL-SUBCONJ ()))))))))))
  (not (if POST (setq NOU T))))

(defun MASCARA (L / REPOST INI OK JJ 9**9 0-**9*9** 0-LLL 0-L)
  (setq LL () SS (ssadd) REPOST POST)
  (if POST
      (progn
        (if (COL->FILA) (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** **9*9** 0-LLL LLL)
```

```
            (progn
              (setq JJ (if (< Y 6) '(1 0 2) '(2 1 0)) 0-LLL (F=3 **9*9**) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**)
                    0-LLL (F=3 LLL) P (list X (rem Y 3))))))
      (setq 0-**9*9** **9*9** 0-LLL LLL))
  (foreach N L1A9
    (if REPOST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "CLIC sobre"
                        "el nombre..."))
    (if (and (member N L)
             (or (not LLL)
                 (if REPOST
                     (progn (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL))
                     (EXPLORA-3*9 0-**9*9** 0-LLL N P)))
        (progn
          (if REPOST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         ((= N 2)     '(0 0.15))
                         ((= N 3)  '(0.15 0.15))
                         ((= N 4) '(-0.15 0))
                         ((= N 5)     '(0 0))
                         ((= N 6)  '(0.15 0))
                         ((= N 7) '(-0.15 -0.15))
                         ((= N 8)     '(0 -0.15))
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if REPOST (TREU-RETOL))))
  (if REPOST
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y))))
  (setq POST (if NOU T REPOST)))
```

Segueixen pendents d'explicació els additaments de **MASCARA** representats pel bloc de codi **(if POST (progn ...))** que s'inicia a la 3ª línia, i anirem de dret a

```
  ..............
  (foreach N L1A9
    (if REPOST (RETOL-2 "Espera't" "al missatge" "" "" "" "" "" "CLIC sobre"
                        "el nombre..."))
    (if (and (member N L)
             (or (not LLL)
                 (if REPOST
                     (progn (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL))
                     (EXPLORA-3*9 0-**9*9** 0-LLL N P))))
        (progn
          (if REPOST (TREU-RETOL))
          (command "TEXTO" "MC" (cond ((= N 1) "-0.15,0.15")
                                      ......................
                                      (   T     "0.15,-0.15")) 0.05 0 (itoa N)
                   "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if REPOST (TREU-RETOL))))
  ..............................
```

perquè en l'expressió condicional

```
    (if (and (member N L)
             (or (not LLL)
                 (if REPOST
                     (progn (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL))
                     (EXPLORA-3*9 0-**9*9** 0-LLL N P)))) ...
```

hi ha el nucli de la disquisició que ens ocuparà una estona. Abans de res, aclarim que la 2ª línia (**(or (not LLL)**...) no té més objecte que garantir la presentació inicial del caseller 3×3, fruit de l'accés **(MASCARA L1A9)** de la funció **TECLAT**, que s'executa a l'inici d'**OMPLE-SUDOKU**, quan **LLL** encara no s'ha creat. Dit això, cal

tornar amb més deteniment al joc entre **EXPLORA-3*9** i **RE-MEMBER** (o, si ho preferiu, entre **POST** i **REPOST**) quan **POST** s'activa per primer cop, perquè sense esmentar-ho hem abandonat les premises que seguíem. I, per fer-ho amb més seguretat, seria bo recapitular:

- **T+D<12** identifica situacions en què, si la pròxima casella que s'emplenés fos P_{i+1} i es fes amb el valor **N**, el sudoku esdevindria inviable. Per aquesta raó, els valors d'assignació a la pròxima casella (encara que aquesta no sigui P_{i+1}) i successives es posen sota el control de **RE-MEMBER**. A aquestes situacions, que anomenem "precatastròfiques", s'hi arriba després d'haver emplenat una casella P_i amb un determinat valor, i **T+D<12** les identifica a **A-9*9** (el nom que pren **9*9** en **ACTUALITZA**) en localitzar-hi alguna de les 6 configuracions típiques que hem anomenat **supòsits**.
- **EXPLORA+1** contempla totes les possibilitats del pròxim emplenament i identifica si n'hi ha alguna (casella P_{i+1} emplenada amb el valor **N**) que, si es realitzés, convertiria el sudoku en inviable. Per aquesta raó, els valors d'assignació a la pròxima casella (encara que aquesta no sigui P_{i+1}) i successives es posen sota el control de **RE-MEMBER**. A aquestes situacions previsibles i que volem evitar, que anomenem "catastròfiques", s'hi arriba després d'haver emplenat una casella P_i amb un determinat valor, i **EXPLORA+1** les identifica entre el conjunt de totes les possibles **A-LLL+1** (el nom que pren en aquesta funció la **LLL** actualitzada a l'emplenament P_{i+1}) en trobar que les seves línies, columnes o requadres 3×3 no respecten les 3 normes establertes.
- **EXPLORA-3*9** contempla totes les possibilitats d'emplenament de la casella P_{i+1} que anem a emplenar i, si troba algun valor d'assignació **N** que convertiria el sudoku en inviable, fa dues coses: 1) impedir aquesta assignació, posant-la fora de l'abast de l'usuari, i 2) deixar l'assignació a la pròxima casella i següents sota el control de **RE-MEMBER**. Com **EXPLORA+1**, **EXPLORA-3*9** identifica aquestes situacions catastòfiques en trobar que les seves línies, columnes o requadres 3×3 no respecten les 3 normes establertes, però en comptes de fer prospeccions a totes les caselles buides es limita a P_{i+1}, que l'usuari ja ha decidit emplenar.

Situades totes dues en **ACTUALITZA**, **T+D<12** i **EXPLORA+1** eren equivalents en la seva actuació: com que l'únic que interessava era endarrerir tot el que fos possible la intervenció de **RE-MEMBER**, quan s'activava **POST** no es prenia nota de quina casella P_{i+1} n'era responsable (de fet, hi podia haver diversos focus potencials -caselles susceptibles d'esguerrar el sudoku si s'emplenaven amb determinats valors-) i **POST** es mantenia activada encara que la casella emplenada a continuació no fos aquesta. Fins i tot, amb **POST** activada i **RE-MEMBER** actuant, hauríem pogut fer que **MASCARA** detectés si algun valor **N** compatible en primera instància (present en l'element de **LLL** corresponent a la casella) havia estat vetat per **RE-MEMBER**, per mantenir **POST** activada i seguir amb **RE-MEMBER** o, en cas contrari, desactivar **POST** i retornar a **T+D<12** o **EXPLORA+1** intervenint des d'**ACTUALITZA**. Però ni volíem complicar més les coses ni teníem gens clar que l'exigència de verificar exhaustivament els valors compatibles amb la viabilitat del sudoku anés lligada a la imminència d'allò que hem anomenat "situacions catastròfiques": deixant-ho així ens curàvem en salut. Vist amb més detall, si **T+D<12**, just després d'emplenar la casella P_i, detecta en la pseudomatriu **9*9** actual una configuració que encaixa en algun dels **6 supòsits** (situació actual precatastròfica), mentre que **EXPLORA+1** ha de fer una prospecció als possibles emplenaments P_{i+1} i detectar en la **LLL** corresponent una vulneració a alguna de les 3 normes (eventual situació catastròfica en l'emplenament següent), la diferència és purament instrumental però les conseqüències són idèntiques: **POST** s'activa després d'un emplenament P_i controlat per **T+D<12** o **EXPLORA+1**, de manera que l'emplenament P_{i+1} i successius ja es faran sota la supervisió de **RE-MEMBER**.

Des de **MASCARA**, **EXPLORA-3*9** hauria de fer el mateix, si fa no fa, però tal com hem articulat el relleu entre **EXPLORA-3*9** i **RE-MEMBER** (la fa poc esmentada divisió de funcions entre **POST** i **REPOST**) el funcionament és diferent i el resultat també pot ser-ho: si, amb P_{i+1} com a casella actual (i **POST** = **REPOST** = **nil**), **EXPLORA-3*9** es troba que un valor **N** compatible en primera instància ja no ho és en segona perquè vulnera la **norma 1**, **2** o **3**, abans que en el següent accés a **MASCARA** s'activi **REPOST** i l'emplenament de la casella P_{i+2} passi a ser supervisat per **RE-MEMBER**, la resta de valors en principi assignables a P_{i+1} seguiran sota el control d'**EXPLORA-3*9**, i estarem jugant amb unes normes que, sense voler entrar a qualificar d'encertades o errònies, són diferents de les que estàvem aplicant: no oblidem que, mentre sigui

REPOST = **nil** i conseqüentment l'exploració vagi a càrrec d'**EXPLORA-3*9**, el que fa **MASCARA** amb la casella P_{i+1} és una prospecció i l'usuari únicament podrà abordar l'assignació si tots els valors 1.. 9 presents a l'element de **LLL** representatiu de la casella passen el test; altrament, si algun d'aquests valors fos vetat, caldria cancel·lar l'exploració en curs, activar **POST** i repetir-la amb **RE-MEMBER**, propòsit per al qual **MASCARA** hauria d'incorporar una variable **REP** per complicar l'expressió **(foreach N L1A9 ...)** precedent amb una repetició que en commutés el valor

```
  ..............
    (repeat 2
    (if (setq REP (not REP))
        (progn
          (foreach N L1A9
            (if REPOST (RETOL-2 "Espera't" "al missatge" "" "" "" "" ""
                                "CLIC sobre" "el nombre..."))
            (if (and (member N L)
                     (or (not LLL)
                         (if REPOST
                             (progn
                               (setq INI T OK ())
                               (RE-MEMBER 0-**9*9** 0-LLL))
                             (if REP
                                 (setq REP (EXPLORA-3*9 0-**9*9** 0-LLL N P))))))
                (progn
                  (if REPOST (TREU-RETOL))
                  (command "TEXTO" "MC" (cond ((= N 1) "-0.15,0.15")
                                              .....................
                                              (   T     "0.15,-0.15"))
                                        0.05 0 (itoa N)
                                 "DESIGNA" (ssadd (entlast) SS) "")
                  (setq LL (cons N LL)))
                (if REPOST (TREU-RETOL))))
          (if (not REP)
              (progn
                (command "BORRA" SS "")
                (setq SS (ssadd) REPOST T)))))))
  ..............................
```

i reinicialitzés el caseller 3×3, esborrant-hi els valors que **EXPLORA-3*9** ja havia tret per pantalla. De seguida veureu que amb això no n'hi ha prou, però abans de seguir convé aclarir que l'exigència que sigui **RE-MEMBER** qui digui la seva obeeix més a la necessitat d'assegurar que els valors 1... 9 a l'abast de l'usuari seran compatibles amb la viabilitat del sudoku que no pas al caràcter "catastròfic" dels valors refusats: quan **EXPLORA-3*9** detecta que l'assignació d'un valor a la casella incompleix alguna de les tres normes podem assegurar que **RE-MEMBER** confirmaria el veredicte de rebuig, però no passa el mateix amb els valors acceptats pel primer d'aquests dispositius, que podrien ser rebutjats pel segon. Aquesta afirmació no és gratuïta sino que l'autor l'ha comprovada en una gran quantitat d'assajos, en què una mateixa seqüència d'emplenaments era realitzada dues vegades, una amb una versió de **MASCARA** on hi intervenia exclusivament **EXPLORA-3*9** i l'altra on ho feia **RE-MEMBER**, prenent nota totes dues vegades dels valors apareguts en el caseller 3×3: en el primer cas sortien els mateixos valors que en el segon, però de vegades algun de més. Aquestes experiències, de retruc van donar lloc a una conclusió molt més trencadora, que no trigarem a desvelar.

De moment (i repetim que sense prejutjar si **EXPLORA-3*9** és més o menys fiable que els seus predecessors), aclarirem perquè l'últim dispositiu, malgrat la correcció, segueix funcionant de forma diferent a les versions que treballaven amb **T+D<12** o amb **EXPLORA+1** des d'**ACTUALITZA**, i no parlem de sistemes alternatius pet realitzar una mateixa feina (com ara les dues funcions esmentades) sinó de feines diferents. Amb **EXPLORA-3*9**, quan **MASCARA** es troba amb una primera situació catastròfica no es pot dir que això de "primera" designi la mateixa situació que quan ho afirmàvem en les versions precedents (potser s'entendrà millor triant **EXPLORA+1** de referent), perquè aleshores definíem com a precatastròfica la primera situació P_i que pogués ser seguida per una de catastròfica P_{i+1}, emplenéssim o no després la casella susceptible de provocar la catàstrofe. Ara, en canvi, només podem rastrejar les situacions catastròfiques lligades a les caselles que emplenem efectivament, raó per la qual és ben possible que **POST** s'activi després d'emplenar una casellla P_j,

per haver detectat una situació catastròfica en anar a emplenar $\mathbf{P_{j+1}}$, acció que en la versió esmenada s'haurà iniciat sota el control d'**EXPLORA-3*9** i s'haurà repetit amb **RE-MEMBER** per seguir així fins al final, quan amb les versions amb **T+D<12** o **EXPLORA+1** potser s'hi hauria arribat abans (en un emplenament $\mathbf{P_i}$, sent **i < j**).

Si no hagués estat per les experiències de què parlàvem línies enrera, després de les consideracions fetes en l'últim paràgraf el dilema estaria servit, perquè en tractar de donar un pas més i escatir si, malgrat no contemplar tantes situacions com la versió amb **T+D<12** (millor que deixem córrer la d'**EXPLORA+1**, per la lentitud que la caracteritza), l'última versió amb **EXPLORA-3*9** era fiable, havíem trobat un contraexemple que demostrava que no era així. Els dos exemples i el contraexemple, que us oferim tot seguit, són d'allò més incorrectes políticament perquè la forma en què s'inspiren és ni més ni menys l'esvàstica nazi, que l'autor volia construir usant sèries de valors correlatius en els braços i que era precís completar amb 8 caselles plenes de més (fins a tenir-ne 25), per tal d'assolir sudokus amb solució única. En tots tres (que mostrem amb les caselles que componen el SUDOKU-PROBLEMA destacades en negreta i la resta en caràcters normals), l'ordre d'emplenament era el mateix: es començava pel requadre central 3×3, amb els valors **1**, **2**, **3**, **4** i **5**, s'omplien les 8 caselles de les cantonades i se seguia amb els 4 braços: superior (**3**, **4** i **5**), dret (**4**, **5** i **6**), inferior (**5**, **6** i **7**), i esquerre (**6**, **7** i **8**).

1 **8** 4	6 **5** 9	3 2 **7**	**2** **4** 8	6 **5** 7	9 3 **1**	**1** **2** 4	6 **5** 8	3 7 **9**
5 2 7	**4** 3 8	1 6 **9**	5 9 7	**4** 3 1	2 6 **8**	5 9 7	**4** 3 2	1 6 **8**
6 9 **3**	7 2 1	**4** 8 5	6 1 **3**	8 9 2	**4** 7 5	6 8 **3**	7 9 1	**4** 2 5
4 6 9	**2** 8 **3**	7 **5** 1	4 6 9	**2** 8 **3**	1 **5** 7	4 1 9	**2** 8 **3**	7 **5** 0
8 3 5	9 **1** 7	2 4 **6**	**8** 5 2	7 **1** 9	3 4 **6**	**8** 0 5	9 **1** 7	2 3 **6**
2 **7** 1	**5** 6 **4**	8 9 3	3 **7** 1	**5** 6 **4**	8 2 9	3 **7** 2	**5** 6 **4**	8 9 1
7 1 **6**	8 9 2	**5** 3 4	7 2 **6**	1 4 8	**5** 9 3	2 3 **6**	8 4 9	**5** 1 7
3 5 8	1 4 **6**	9 7 2	**9** 8 5	3 2 **6**	7 1 4	**7** 5 8	1 2 **6**	9 4 3
9 4 2	3 **7** 5	6 **1** **8**	**1** 3 4	9 **7** 5	6 **8** **2**	**9** 4 1	3 **7** 5	6 **8** **2**

Doncs bé, si us heu fixat en els dos zeros que hi ha en el sudoku de la dreta, no és cap error, sinó que hi són per representar caselles buides, com havíem fet fins ara. Però, ¿que no estàvem parlant de sudokus estrictes, amb la seva solució?: sí, pel que fa als dos primers, però el tercer és un intent fallit en què el conflicte es concentra a les dues últimes caselles emplenades, que són 2,4 (**7**) i 1,5 (**8**). La raó de tot havia estat la intuició que un símbol tan sinistre pel seu significat històric com fascinant com a disseny gràfic havia d'encaixar bé en el joc que ens ocupa, usant successions numèriques simples. Atès que amb les 17 caselles ocupades per l'esvàstica el sudoku no quedava definit, el primer intent va ser emplenar les caselles de les cantonades (21). Com que amb una a cada cantó no n'hi havia prou, es va intentar amb dues (25): la segona és veïna de la primera, no té contacte amb els braços i està situada de manera que la figura manté la simetria central. Els dos primers exemples responen a l'intent, però es buscava millorar algun aspecte relatiu a l'ordenació numèrica o bé reequilibrar la presència dels enters 1... 9 per veure si així podíem baixar a 23 caselles. Precisament va ser fent proves amb la configuració de la dreta quan l'autor es va adonar que alguna cosa no marxava: seguint l'ordre d'emplenament anunciat, la versió amb **T+D<12** només oferia 3 valors per a la casella 1,5 (**2**, **3** i **4**), mentre que la versió amb **EXPLORA-3*9** n'oferia dos més (**5** i **8**); de fet, ja a la casella precedent, 2,4 (**1**, **3**, **6**, **7** i **9**), **EXPLORA-3*9** n'oferia un més (**8**). Aquesta última versió no oferia garanties, sens dubte perquè el valor **8** a 1,5 era incompatible amb la viabilitat del sudoku, i n'era la prova que, si l'introduïem, **ACTUALITZA-PLUS** desencadenava l'emplenament de la resta de caselles... tret de les esmentades 2,5 i 9,6, que quedaven buides, i si preteníem omplir-les el caseller 3×3 ens mostrava que no hi havia candidats. L'alarma **POST** no s'havia disparat, rellevant la funció **EXPLORA-3*9** i passant el comandament a **RE-MEMBER**, i n'era la prova que s'arribava a aquest galdós final sense que hagués aparegut l'avís *Espera't al missatge "CLIC sobre el nombre..."*.

Per confirmar el diagnòstic que expresàvem línies enrera i localitzar sobre aquest contraexemple el moment exacte en què **EXPLORA-3*9** hauria d'haver activat **POST** i no ho va fer, haurem de rastrejar aquest mecanisme a la versió amb **T+D<12**. Utilitzant aquesta versió, **POST** s'activa després d'haver emplenat la casella 4,8 (**4**) i donar-se el **supòsit 1** (6 caselles plenes en el requadre 9×3 superior, totes amb un valor diferent). Ja havíem dit que aquest supòsit, "... a més de respondre a situacions

concretes ... també servirà per limitar el marc general d'actuació i fer possible que la formulació dels supòsits següents sigui relativament senzilla". I aquest és un dels casos en què es limitava a exercir un paper cautelar, perquè la situació no registrava cap altre supòsit, raó per la qual, si en lloc d'usar aquesta versió s'hagués arribat al mateix punt utilitzant la que arrossega feixugament **EXPLORA+1**, **POST** hagués seguit desactivat, i hauríem hagut d'emplenar 2 caselles més, 5,9 (**5**) i 7,7 (**4**), i assistir a l'emplenament automàtic (**ACTUALITZA-PLUS**) d'una altra més, 3,9 (**4**), perquè s'activés i **RE-MEMBER** assumís el control dels valors assignables. Ara es presentaria el **supòsit 2**, centrat altra vegada en el requadre 9×3 superior: observeu **1**, **2** i 4 omplint el primer tercet de la 9ª fila, la casella 9,8 (**8**) i la possibilitat que l'emplenament següent es fes a qualsevol casella del segon tercet de la 7ª fila, amb el valor **8**; aquesta possibilitat (i en concret, l'emplenament de la primera casella d'aquest tercet, 4,7, que visitaria en primer lloc) seria la registrada per **EXPLORA-3*9** < **EXPLORA+1** < **ACTUALITZA** i activaria **POST** per infracció de les **normes 1** i **2**. La detecció es produiria en el requadre 3×3 superior esquerre de **A-9*9+1** (esquerra) i **A-LLL+1** (dreta), on podreu apreciar l'absència del valor **8** i la presència de 4 valors en les 3 llistes (0 0 0 0 **5 6 7** 0 **9**):

1	**2**	**4**	0	0	0
(0 7)	(1 7)	(2 7)	(0 0 0 0 **5 6** 0 0 0)	(0 0 0 0 **5 6 7** 0 **9**)	(0 0 0 0 **5 6 7** 0 **9**)
(0 6)	(1 6)	**3**	(0 0 0 0 **5 6** 0 0 0)	(0 0 0 0 **5 6 7** 0 **9**)	0

Com que la versió d'**EXPLORA-3*9** < **MASCARA** no verificarà l'emplenament hipotètic de la casella 4,7 i les seves repercussions arreu, sinó que es limitarà a la casella següent, 8,7 (**5**), vet aquí el senyal de perill que ha estat incapaç de reconèixer i que set caselles endavant desembocarà en un carreró sense sortida.

La trista conclusió és que el mètode que podíem haver anomenat "de les 3 normes", que si no és un sistema complet i sense llacunes almenys en té tot l'aspecte, sols és fiable en prospeccions avançades una casella, i així resulta inacceptable per l'extremada lentitud. D'altra banda, el teòricament equivalent "dels 6 supòsits" funciona a una velocitat raonable però la formulació teòrica no té la contundència de l'anterior: no és més que la col·lecció de situacions crítiques identificades fins ara, sense cap garantia que no n'hi pugui haver més. Però ambdós dispositius, paradoxalment, no són els que havien d'abordar el tema principal proporcionant-nos la certesa sobre els valors que podíem assignar a la casella actual sense arruinar el sudoku, sinó mers corifeus. Usant un símil evangèlic, podríem dir que **T+D<12** o **EXPLORA+1** eren el Joan Baptista que clamava en el desert per anunciar la vinguda de l'autèntic Redemptor **RE-MEMBER** i aixecar acta de la seva presència, realitzant mentrestant una tasca de neteja rutinària. Però, ¿per què dimonis necessitàvem un corifeu? ¿Que no hagués pogut intervenir **RE-MEMBER** de bon principi? Doncs sí, però el ruc de l'autor (que, convicte i confés, ara prescindeix del plural de modèstia) va cometre un pecat d'aquells que no té perdó: donar per fet que una presumpció és verdadera sense haver-se pres la molèstia de comprovar-la (naturalment, parlem de fenòmens comprovables), tan fàcil com hagués estat verificar si el comportament de **RE-MEMBER** era com l'autor pressuposava, sobretot abans d'embarcar-se inútilment en els afers que ocupen les últimes vint-i-tantes pagines. Els prejudicis de l'autor al respecte queden ben palesos en dues cites, una que correspon a les reflexions prèvies als 6 enunciats que conformen el dispositiu **T+D<12** i l'altra, posterior, que reflecteix la insatisfacció que duria a substituir-lo per **EXPLORA+1**:

> "Igual com d'aquí a sis capítols (*Condicions mínimes*), quan ja es disposi d'una SOLUCIÓ, CREADA, COPIADA o CANÒNICA, i haguem començat tranquil·lament el procés d'activació de caselles (emplenament amb el valor predeterminat per la solució), en constatar que es donen les condicions mínimes perquè el conjunt de caselles emplenades pugui ser un SUDOKU-PROBLEMA en tota regla s'activarà l'alarma **POST** per realitzar l'inventari de les solucions compatibles, inventari que portarà el seu temps i que convé ajornar tot el que sigui possible, també a **OMPLE-SUDOKU** farem **POST** = **T** per caracteritzar el moment a partir del qual la presentació dels valors per omplir una casella ha d'anar més enllà d'una aplicació primària de les regles del joc."

> "Potser calia cercar un procediment més arrelat en els fonaments teòrics del procés i deixar clares algunes qüestions que acceptàvem implícitament però sense ser massa conscients dels seus límits:

Que hi havia un remei infal·lible però que en la pràctica no semblava assumible: fer **POST** = **T** d'entrada (és a dir, usar **RE-MEMBER** de bon començament). Perquè si ens estem trencant les banyes per fabricar un dispositiu que permeti detectar a temps situacions que durien a un carreró sense sortida és precisament per evitar que **RE-MEMBER** hagi d'actuar en un escaquer pràcticament buit, circumstància que es traduiria en llarguíssims temps d'espera, potser no els primers (que tindrien molts graus de llibertat però pocs condicionaments) però sí amb poques caselles plenes (hauríen baixat els primers però pujarien més ràpidament els segons)."

Els subratllats són d'ara però posen en evidència la desorientació inicial. Potser sota la influència d'haver començat a treballar en l'obtenció del SUDOKU-PROBLEMA a partir de la SUDOKU-SOLUCIÓ, sense preocupar-li massa la procedència de l'últim (l'autor no va pensar en la pertinència d'incorporar una opció A -SOLUCIÓ CREADA- fins que no va tenir resolta aquesta qüestió), ja que en aquell context sí que té sentit parlar d'unes condicions mínimes, amb un abans i un després, va abordar el disseny de l'opció A amb la idea preconcebuda que també en la tasca d'improvisar la solució, emplenant l'escaquer 9×9 casella a casella sota control del programa, hi havia d'haver algun llindar o, amb les paraules de la primera cita reproduïda, un moment a partir del qual la presentació dels valors per omplir una casella ha d'anar més enllà d'una aplicació primària de les regles del joc. A mesura que es perfilaven les funcions **MASCARA** i **RE-MEMBER** s'ha anat veient que aquesta divisió en dues etapes no era res consubstancial al procés sinó una conveniència pràctica, i la gran badada ha estat no pensar que podia ser pitjor el remei que la malaltia. Argumentar que abans d'avaluar el remei calia preparar-lo seria una excusa de mal pagador, perquè el que calia abans d'això era veure si la malaltia era tan greu: és imperdonable que, sense més base que unes quantes proves molt localitzades en els pocs exemples utilitzats per a la posta a punt de **RE-MEMBER**, l'autor tingués la gosadía de fer les afirmacions subratllades en la segona cita.

Però la veritat acaba obrint-se pas tard o d'hora, per més que vulguem ignorar-la: ultra no estar a l'altura de la missió que se li encomanava, ja hem comentat que els temps d'espera amb **EXPLORA-3*9** eren més regulars però més llargs que els de **RE-MEMBER** (si exceptuem esporàdiques sortides de mare en aquests, que més endavant abordarem), tot i que sempre es podia argumentar que els primers es produïen en un escaquer encara molt buit mentre que els segons ho feien en un força més poblat. Ara bé: quan, per poder prendre nota dels valors autoritzats per **MASCARA** i del temps que trigaven a aparèixer en el caseller 3×3, seguint una mateixa seqüència d'emplenament de caselles, es van improvisar dues versions experimentals en què **EXPLORA-3*9** i **RE-MEMBER** representaven papers exclusius i excloents, aquest últim bastió argumental per negar l'innegable es va ensorrar. Veiem aquestes versions.

En totes dues **POST** va ser inicialitzat a **T** al començament d'**OMPLE-SUDOKU**

```
(defun OMPLE-SUDOKU (/ CURSOR P-CURS C->F)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq **9*9** (INI-COORDS)
              P-CURS '(-1.1919 0.4419)
              POST T)
        (INI-LLL)
        (while (not (COMPLET-9*9 **9*9**)) ...)
        ...))
  ...)
```

i la funció **ACTUALITZA** va quedar reduïda a la mínima expressió

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if REAL (ACTUALITZA-PLUS))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))
```

Pel que fa a **MASCARA**, en una es va mantenir la variable **REPOST**, per no haver-nos de molestar definint en **EXPLORA-3*9** una variable local *ad hoc* que substituïs **POST**,

```
(defun MASCARA (L / REPOST INI OK JJ 9**9 0-**9*9** 0-LLL 0-L)
  (setq LL () SS (ssadd) REPOST POST)
  (if POST
      (progn
        (if (COL->FILA) (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** **9*9** 0-LLL LLL)
            (progn ...))))
  (foreach N L1A9
    (if REPOST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                        "el nombre...\""))
    (if (and (member N L)
             (or (not REPOST) (setq POST ()) (EXPLORA-3*9 0-**9*9** 0-LLL N P)))
        (progn
          (if REPOST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         .......................
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if REPOST (TREU-RETOL))))
  (if (setq POST REPOST)
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))
```

mentre que l'altra es va deixar així

```
(defun MASCARA (L / INI OK JJ 9**9 0-**9*9** 0-LLL 0-L)
  (setq LL () SS (ssadd))
  (if POST
      (progn
        (if (COL->FILA) (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** **9*9** 0-LLL LLL)
            (progn ...))))
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (if POST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         .......................
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if POST (TREU-RETOL))))
  (if POST
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))
```

A més de confirmar-nos que la primera no era bona, per oferir-nos valors de massa (que condemnaven el sudoku a quedar blocat, si eren assignats a una casella), la segona demostrava ser força més ràpida, sobretot al principi, cosa que desmuntava la justificació d'una estratègia de dues etapes dependents de **POST**, no només per innecessària sinó per contraproduent, i consagrava la intervenció de **RE-MEMBER** en solitari com el millor i definitiu procediment. Dit això, algú pensarà que estava bé que construïssim aquestes dues versions a partir de les precedents de la forma més expeditiva possible, però que havent arribat a la conclusió que la segona ja és una base prou ferma per seguir endavant, podríem depurar-la d'additaments tan superflus com **POST**, que realment ja no hi pinta res (per a la presentació inicial

del caseller central 9×9 ple, no costaria massa de trobar un sistema alternatiu a les crides efectuada des de **TECLAT** i **2-9V**, encara en les condicions **POST** = **nil**): prescindir d'aquesta variable en tot l'àmbit d'**OMPLE-SUDOKU** evitaria activar-la i permetria esborrar-ne les nombroses referències que té **MASCARA**, alliberant totes les expressions que hi estaven condicionades i simplificant la sentència

```
(if (and (member N L)
         (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL))) ...)
```

que quedaria reduïda a

```
(if (and (member N L)
         (or (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL))) ...)
```

Això fóra correctíssim, però ja que l'autor no us ha estalviat el càstig de seguir una tediosa història que no duia enlloc i que sobretot era la història d'un fracàs (en descàrrec seu només afegirà que no ho ha fet amb intenció dolosa, sinó guiat per la confiança que algú, algun dia, pugui treure algun profit dels 6 supòsits o de les 3 normes) en compensació us mostrarà una drecera: efectivament, ara per ara **POST** s'ha quedat sense paper a l'obra que es representa, però més endavant (en el capítol *Condicions mínimes*) en jugarà un de similar al que fins fa poc preteníem donar-li, aquí sense que hi intervingui la funció **MASCARA**; però és que encara més endavant (en el capítol *Un nou camí i també una drecera*) **POST** tornarà a treure el cap i hi haurà un accés a **MASCARA** (tot i ser virtual la seva intervenció, en el sentit que el caseller 3×3 no surt en pantalla), on **POST** pot facilitar les coses. Tot i així, és millor posposar la decisió i provisionalment deixar-ho com estava.

Per si algú es pensava que amb la coronació de **RE-MEMBER** com a àrbitre absolut de l'oferta que es produeix a **MASCARA** quedava tot resolt, a banda de felicitar-lo per l'optimisme de pedra picada, li haurem de recordar el que des de la primera versió de **MASCARA** hem anat repetint: que encara quedaven per justificar certs additaments d'aquesta funció. I, com que ja és l'hora de posar-nos-hi, començarem resumint les característiques de les dues antagonistes:

- Respecte a **EXPLORA-3*9**:
  - Que l'encerta quan dictamina que algun valor és incompatible, però que de tant en tant falla en presentar com a compatibles valors que, si s'assignessin a la casella actual condemnarien el sudoku.
  - El temps que triguen a aparèixer en el caseller 3×3 els valors presentats com a compatibles són sensibles però regulars, tendint a escurçar-se a mesura que l'escaquer es va omplint.
- Respecte a **RE-MEMBER**:
  - Que no falla mai: els valors presentats com a compatibles sempre donaran lloc a una solució, si més no.
  - El temps que triguen a aparèixer en el caseller 3×3 els valors presentats com a compatibles són irregulars però, de mitjana, inferiors als d'**EXPLORA-3*9**. Però de tant en tant es produeixen temps d'espera tan elevats que l'usuari pot tenir la sensació que l'execució s'ha col·lapsat o ha entrat en cercle viciós.

L'inconvenient pràctic més greu associat a **RE-MEMBER** es la seva imprevisibilitat i no pas la lentitud, intrínsecament associada a determinats estadis del procés, que l'autor li atribuïa. Tant de bo les coses anessin com aquest es pensava!: que el procés era tan ràpid en emplenar la primera dotzena de caselles com en l'última, i que la llarga etapa intermèdia era la que tenia un comportament més erràtic, amb preemplenaments inesperadament dilatats. Dissortadament, aquestes sobtades i de vegades inacabables pauses (poden durar hores) també poden produir-se omplint les primeres caselles, reconeixement que no implica impugnar les últimes conclusions i tornar a les hipòtesis inicials (per allò que més val fer front a pauses llargues però gairebé constants que no pas disfrutar de respostes immediates però anar amb l'ai al cor per la por de quedar penjat en qualsevol moment) sinó posar l'acció de **RE-MEMBER** sota algun control.

Per tant, no li donem més voltes a **EXPLORA-3*9**, mirant de treure rendiment a una eina que ha costat de posar a punt però que al capdavall s'ha revelat tan inútil, perquè només és fiable quan el valor **N**, un cop superat el filtre **(member N L)** en les iteracions que tenen lloc a **MASCARA**, és refusat com a candidat; ja hem vist que quan el declara compatible no ho és gens, de fiable, i que en aquest cas només **RE-MEMBER** pot emetre un veredicte encertat. Diem això perquè la temptació d'una sortida eclèctica, consistent en aplegar **EXPLORA-3*9** i **RE-MEMBER** per quedar-nos amb allò que té de millor cada funció, és massa gran. L'autor tampoc no gosaria qualificar de forassenyada aquesta idea, i l'únic que pretén és que ningú no se la prengui com un talismà infal·lible que sempre ens protegirà de **RE-MEMBER** en les seves sortides de mare. Garbellar els candidats pel doble filtre d'**EXPLORA-3*9** i

RE-MEMBER de vegades donarà bons resultats (en el sentit de sostreure el control de la selecció a la segona funció, transferint-lo a la primera però amb garanties) i de vegades no, raó per la qual no entenem que aquest dispositiu mixt sigui una alternativa al sistema recomanat sinó més aviat un recurs *in extremis* reservat per als casos en què aquest ens hagi ficat en una espera que preferiríem evitar ni que fos tornant a començar el procés d'emplenament (en el benentés que l'intent podria no reeixir i hauríem de suportar la llarga espera malgrat tot). Entendrem millor aquesta limitació a la vista del codi en què es tradueix la voluntat expressada. Com que el caràcter de pegat per a casos d'emergència no justifica depurar-lo fins a les últimes conseqüències, ens limitarem a suggerir una substitució puntual en el cos de la funció **MASCARA** (i així fins i tot podríem integrar la variant com un comentari, que passaria a ser codi avaluable traslladant a l'expressió alternativa el caràcter ";"). N'hi haurà prou a substituir

```
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn ...)
```

per

```
    (if (and (member N L)
             (or (not POST) (if (EXPLORA-3*9 0-**9*9** 0-LLL N P)
                                (progn (setq POST T INI T OK ())
                                       (RE-MEMBER 0-**9*9** 0-LLL)))))
        (progn ...)
```

(naturalment, caldrà incorporar-hi les definicions d'**EXPLORA-3*9** i **EXPL-SUBCONJ**). El canvi respon al que dèiem unes línies enrera: si el resultat d'**EXPLORA-3*9** és **T** (**POST**, inicialitzat a **nil** a l'inici d'aquesta funció, es manté així, cosa que en principi implicaria que **N** és compatible) no ens ho creurem i anirem a **RE-MEMBER**; si és **nil** (**POST** s'ha activat, cosa que revela la incompatibilitat de **N**) el donarem per bo. Ara bé, ¿per què dèiem que aquesta híbridació unes vegades farà bon servei i unes altres no?: doncs perquè, contra allò que pugui semblar a primera vista (que **RE-MEMBER** només fa perdre la paciència a l'usuari en exploracions que acaben amb resultat negatiu), els temps d'execució llargs de **RE-MEMBER** tant poden donar-se en casos de compatibilitat (i la intervenció d'**EXPLORA-3*9** no ens n'alliberarà, perquè el veredicte haurà de ser refrendat per **RE-MEMBER**, amb una exploració que pot ser curta o llarga) com d'incompatibilitat (en què la primera intervenció serà concloent i farà innecessària una segona, que igualment podria haver estat curta o llarga). En conclusió, encara que hi hagi sort i puguem eludir algunes o totes les intervencions dilatades de **RE-MEMBER**, pel cap baix sempre haurem de suportar els temps d'espera d'**EXPLORA-3*9** (això amb els valors refusats per aquesta funció, perquè quan hi hagi presumpció de compatibilitat s'hi sumaran els de **RE-MEMBER**).

Etapa prèvia: crear una solució, III (roda el món i torna al Born)

Tornant a la necessitat de disposar d'un procediment per "domesticar" **RE-MEMBER**, agilitzant l'exploració que determina l'aptitud dels candidats per a l'emplenament de cada casella, almenys en la mesura suficient per evitar les llargues esperes a què ens referíem, cal confessar que disposem d'una eina prou versàtil però que les situacions que es poden presentar són tan diverses i complexes que no serà fàcil utilitzar-la encertadament. Aquesta eina no és sinó la possibilitat de modificar l'escaquer 9×9 amb algunes de les 1.218.998.108.160 permutacions descrites en el capítol *Tornem a començar: de la solució al problema*, per tal que a la variant obtinguda (on la casella actual ocuparà generalment una altra posició) no calgui donar tants passos (autorecursions de **RE-MEMBER**): poden usar les que en el menú del programa hem batejat com a opcions F1, F3, FS, C1, C3, CS i TR; amb F2 i C2 ens complicaríem la vida, i PV no serviria de res. Naturalment, si transformant l'escaquer n'afectem l'exploració és perquè aquesta segueix una pauta posicional: sempre recorre les files d'esquerra a dreta i avança per les files de baix a dalt; altrament, intercanviar el contingut de les caselles no serviria de res. Doncs bé, els additaments de **MASCARA** als quals ens veníem referint de lluny són el primer i més rudimentari intent d'aplicar aquest principi, i consten de dues expressions: 1) la que depèn de la sentència condicional situada a la 3ª línea

```
  (if POST
      (progn
        (if (COL->FILA) (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** **9*9** 0-LLL LLL)
            (progn
              (setq JJ (if (< Y 6) '(1 0 2) '(2 1 0))
                    0-LLL (F=3 **9*9**) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**)
                    0-LLL (F=3 LLL)
                    P (list X (rem Y 3)))))))
```

que té per objecte aplicar a ****9*9****, a **LLL** i a **P** la transposició TR, si es donen determinades circumstàncies, i permutacions F3 entre requadres 9×3, si se'n donen d'altres (en aquestes, ****9*9**** i **LLL** passen a ser anomenats **0-**9*9**** i **0-LLL**, i així seran adoptats com arguments de **RE-MEMBER**); 2) la que depèn de la condicional final i retorna les coses al seu lloc

```
  (if POST
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ()))
        (setq P (list X Y))))
```

No reproduïm aquí les funcions **COL-FILA** i **TRANSPOSA-HO** (que ja teniu a la primera versió en què sortia **RE-MEMBER**) ni **F=3** (vehicle de les permutacions F3, que tres capítols enrera servia **CAN->VAR**), però n'explicarem l'actuació: **COL-FILA** explora els requadres 9×3 i 3×9 que contenen la casella actual **P**, comptant les caselles plenes de cada un i resultant **T** o **nil** segons que en tingui més el 3×9 o el 9×3, valor que s'assigna a la variable **C->F**; **TRANSPOSA-HO**, com el seu nom suggereix, obté les transposades de ****9*9**** i **LLL**, permutant les coordenades dels elements del primer que corresponen a les caselles buides, i també les de **P**. D'altra banda, transposades o no aquestes variables, fixeu-vos que la permutació F3 sempre actua deixant en posició inferior el requadre 9×3 on se situa **P**. Entesos, pero tot això per què? Doncs perquè, conscient que era tan difícil conèixer *a priori* el camí que havia de recórrer **RE-MEMBER** (la llargada del qual, comptant avanços i retrocessos, determina la durada de l'exploració) com el seu resultat (si existís algun sistema de predicció, l'usaríem en lloc de **RE-MEMBER**), l'autor creia que al màxim que es podia aspirar era a unes configuracions que, en general (en l'accepció estadística del terme), milloressin els temps de processat. I, com a primeres intuicions, li semblava que el melic de la qüestió era la casella **P** que anàvem a emplenar, en un doble sentit:

- Si no tots, molts dels estancaments del procés (oscil·lacions en el nivell de recursió de **RE-MEMBER** o, si ho veieu més clar imaginant avenços i retrocessos a l'escaquer, les anades i tornades entre dues posicions, sense passar més enllà) estaven relacionats amb les situacions que qualificàvem de "precatastròfiques" (recordeu **T+D<12** i els 6 supòsits, que era on se situava el front de lluita quan a la reraguarda l'autor reflexionava sobre el tema que ara ens ocupa), i algunes d'aquestes situacions s'inscrivien en un requadre 9×3 o 3×9 (**supòsits 1**, **2** i **3**) o s'hi circumscrivien (**supòsits 4**, **5** i **6**), si amb el mot volem dir que només una petita part de les caselles en conflicte se n'escapava. ¿Podria ser que, quan la casella més avançada estigués en el marc del conflicte, el procés es ralentitzés perquè els retrocessos se succeirien sense parar? La hipòtesi era prou raonable, i si l'acceptàvem una cosa era segura: quan el rectangle conflictiu fos apaïsat (requadre 9×3) aquesta etapa accidentada es limitaria a tres línies, mentre que quan fos vertical (requadre 3×9) se n'estendria a nou (a tota l'exploració), les ziga-zagues tindrien més amplitud i la marrada comportaria perdre més temps.
- Deixar el rectangle potencialment conflictiu en posició horitzontal, transposant l'escaquer 9×9 si calia, era la conclusió immediata del que acabem de dir, però queda per veure què hi guanyàvem situant aquest requadre 9×3 a la part inferior. Per copsar-ne la intenció hem d'adonar-nos que cada retrocés és l'abandonament d'una possibilitat inviable (per no tenir una casella cap valor assignable en primera instància) que, quan abans es produeixi, millor. I, considerant que en aquest requadre 9×3 probablement és on més abandonaments s'esdevindrien, no cal insistir en la transcendència de la seva ubicació a l'escaquer, a efectes de la durada de l'exploració: si era el mitjà o el superior, cada retrocés en arribar a alguna casella del requadre podia produir-se després d'un recorregut llarg i el retorn a posicions precedents deixar-nos força enrera; si era l'inferior, ni el recorregut previ ni la pèrdua de posicions no podien afectar més de 3 línies.

Aquest procediment ha anat relativament bé quan la distribució de caselles plenes era més o menys homogènia. Així, mentre l'usuari anés emplenant l'escaquer evitant concentrar-se en determinades àrees, la cosa funcionava sense sobresalts. Però si, per exemple, copiava un SUDOKU-PROBLEMA amb la intenció d'utilizar el programa per obtenir-ne la SUDOKU-SOLUCIÓ, i ho feia avançant línia a línia, de baix a dalt o a l'inrevés, l'acció combinada de transposició i permutació de requadres 9×3 no sols era inútil sinó sovint contraproduent. Per aquesta raó, la intenció primitiva de millorar l'eina, afegint a l'eventual transposició TR i a una permutació en bloc de requadres 9×3 una permutació de files F1 dintre del requadre 9×3 inferior per deixar la casella actual sobre la 1ª fila, una permutació C3 de requadres 3×9 per situar-la en el primer tercet i una permutació de columnes C1 perquè coincidís amb la posició 1,1, s'ha abandonat, en vista dels magres resultats obtinguts en casos en què el primer pas ja ens allunyava de l'objectiu en comptes d'aproximar-nos-hi. Però serà millor veure-ho en un cas concret i, ja que abans havíem jugat amb una esvàstica, canviarem de registre perquè ningú no tregui falses conclusions dels exemples usats. Ara li arriba el torn a la falç i el martell del "socialisme real" (les tendències comunistes opositores, trotsquistes o no, han utilitzat normalment la versió simètrica respecte a un eix vertical), a la qual ha calgut afegir una casella plena de més (a l'angle superior esquerre, lloc sovint ocupat per l'estel de cinc puntes) perquè el sudoku quedés unívocament determinat (aquest pegat no és tan postís si considerem que els quatre valors en posició perifèrica evoquen l'any 1917). Abans d'anar per feina, una última precisió: amb la contraposició dels dos símbols no es pretén donar corda a cert corrent del pensament liberal-conservador que sosté la identitat essencial d'ambdós règims totalitaris: més enllà dels crims del nazisme i de l'estalinisme (la qüestió no és quin va provocar més morts) hi ha el fet que els primers van ser la conseqüència lògica d'una doctrina, mentre que els segons es van cometre en contra dels principis que nominalment deien defensar. Entrant ja en matèria i per evitat que el lector es faci un embolic a causa de la tendència exagerada a aprofitar espai aplegant il·lustracions que corresponen a exemples diferents, direm que en la primera renglera a l'esquerra teniu la SUDOKU-SOLUCIÓ amb les caselles que componen el SUDOKU-PROBLEMA destacades en negreta; al mig, la part del segon que correspon a un emplenament ordenat d'esquerra a dreta i de baix a dalt fins arribar a les caselles 7,4 (**3**) i 8,4 (**6**), sense cap alteració; a la dreta, la mateixa situació però visualitzant les caselles tal i com queden emmagatzemades a **0-**9*9****, sense transposar l'escaquer però permutant el primer i el segon requadre 9×3. En la segona renglera a l'esquerra es representa la mateixa situació però amb les caselles tal i com queden emmagatzemades a **0-**9*9**** després de transposar l'escaquer però però sense cap permutació de requadres; al mig teniu la mateixa situació però visualitzant les caselles tal i com queden emmagatzemades a **0-**9*9**** transposant l'escaquer i permutant el primer i el tercer requadre 9×3.

```
9 6 8   2 3 1   4 7 5       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
3 5 7   6 4 8   2 1 9       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
2 1 4   5 7 9   8 3 6       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0

8 3 2   9 6 5   7 4 1       0 0 0   0 0 0   0 0 0       0 0 3   0 0 2   6 0 0
6 7 1   3 8 4   9 5 2       0 0 0   0 0 0   0 0 0       0 2 0   4 5 6   0 8 0
4 9 5   1 2 7   3 6 8       0 0 0   0 0 7   3 6 0       1 0 0   0 0 0   0 0 7

5 8 3   7 1 2   6 9 4       0 0 3   0 0 2   6 0 0       0 0 0   0 0 0   0 0 0
7 2 9   4 5 6   1 8 3       0 2 0   4 5 6   0 8 0       0 0 0   0 0 0   0 0 0
1 4 6   8 9 3   5 2 7       1 0 0   0 0 0   0 0 7       0 0 0   0 0 7   3 6 0



7 0 0   0 0 0   0 0 0       0 0 3   0 0 0   0 0 0       9 0 0   0 0 1   0 0 0
0 8 0   6 0 0   0 0 0       0 2 0   0 0 0   0 0 0       0 0 0   0 0 0   2 0 0
0 0 6   3 0 0   0 0 0       1 0 0   0 0 0   0 0 0       0 0 4   5 0 0   0 3 0

0 6 2   7 0 0   0 0 0       0 6 2   7 0 0   0 0 0       0 0 0   0 0 0   0 0 0
0 5 0   0 0 0   0 0 0       0 5 0   0 0 0   0 0 0       0 0 0   0 8 0   0 0 0
0 4 0   0 0 0   0 0 0       0 4 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0

0 0 3   0 0 0   0 0 0       7 0 0   0 0 0   0 0 0       9 0 0   0 0 1   0 0 0
0 2 0   0 0 0   0 0 0       0 8 0   6 0 0   0 0 0       0 0 0   0 0 0   2 0 0
1 0 0   0 0 0   0 0 0       0 0 6   3 0 0   0 0 0       0 0 4   5 0 0   0 3 0
```

Doncs bé: els emplenaments representats en la primera renglera, al mig (esborrant la línia **(if (COL->FILA) (TRANSPOSA-HO))** i substituint la següent, **(if (< Y 3)...**, per **(if (< Y 9)...**) i a la dreta (esborrant **(if (COL->FILA) (TRANSPOSA-HO))** però deixant **(if (< Y 3)...**), van com una seda; a l'emplenament esquerre de la segona renglera (deixant la transposició però sense permutar requadres) tampoc hi ha res de nou, pero en el del mig (deixant transposició i permutacions) es produeix una pausa llarguíssima a les caselles 7,4 (**3**) i 8,4 (**6**), que a la representació ocupen les posicions 4,1 i 4,2. I, amb el pudorós qualificatiu "llarguíssima", l'autor vol significar que passen de l'hora. En aquest i altres casos esdevinguts tot fent proves, era com si la feina de **RE-MEMBER** s'alleugés en començar per baix i seguir cap amunt sense interferències del programa en la localització, i encara millor si la densitat d'ocupació es reduïa progressivament, tal i com s'estratifiquen tres líquids no solubles barrejats en un got. Però això no ho podíem imposar, perquè la iniciativa de l'emplenament corresponia l'usuari... tret que aquest exercís el seu arbitri però internament el programa li esmenés la plana, organitzant **0-**9*9**** en funció de la densitat d'emplenament, per requadres 9×3. És clar que això tenia una pega: si l'usuari anava per una banda (elecció de la casella actual) i el programa per una altra (organització dels requadres 9×3 segons les caselles ja ocupades, ni que fos comptant l'actual com una més), podia esdevenir-se que la casella actual fos desplaçada cap amunt. En resum: ¿quin calia prioritzar, a l'hora de arrossegar cap avall requadres 9×3? ¿aquell on l'usuari hagués activat una casella o aquell que en tingués més de plenes?

Le resposta correcta és la segona, i és que abans havíem mantingut una ambigüitat que ens duia a confondre requadre més poblat amb requadre on podien aparèixer més situacions precatastròfiques (desenterrant uns **supòsits** tipològics que en el marc de **RE-MEMBER** ja no significaven res), i mitjançant aquesta assimilació impròpia posàvem en el mateix calaix casella actual i casella terminal (abandonament d'una exploració fallida). El pitjor és que ens complicàvem la vida, perque la realitat és força més senzilla. Començarem mostrant com la posició de la casella actual es gairebé irrellevant.

Amb cadascun dels valors que calgui assajar en la casella actual, l'itinerari seguit per **RE-MEMBER** serà el mateix que si la casella actual fos qualsevol altra de les ocupades i estiguéssim assajant el seu valor. Només hi ha una diferència: que el valor que estem assajant a la casella actual pot ser que sigui incompatible (i per arribar a aquesta conclusió caldrà haver temptejat totes les combinacions possibles amb les caselles buides), mentre que el valor de les altres caselles ocupades ja està comprovat que té solució... o, més aviat, que en tenia, perquè, sota la hipòtesi que la casella actual i el valor que s'assaja ja són una casella consolidada, podria ser que el valor de la casella consolidada que ara fem passar

per actual fos incompatible. Tret d'aquest matís (que l'autor, honestament, no sap fins a quin punt pot desbaratar l'eficiència de procediments que es derivin de les presents reflexions), es pot treure una conclusió de l'afirmació que el precedeix: que sigui una o altra la transformació (obtinguda mitjançant permutacions de tipus F1, F3, FS, C1, C3, CS i TR) que optimitzi el temps d'execució de **RE-MEMBER**, només depèn de la situació de l'escaquer, incloent-hi la casella actual amb el valor que s'assaja, però sense que la posició d'aquesta casella hi influeixi més enllà de la seva presència entre les altres.

Pel que fa a organitzar els requadres 9×3 o 3×9 a partir d'aquell que tingui més caselles plenes, deixant-los en posició apaïsada i ordenant-los de més a menys en sentit ascendent, no té altre objecte que adaptar-se al full de ruta de **RE-MEMBER**. Quantes més caselles ocupades tingui un requadre 9×3, més contribuirà a escurçar el procés: el nombre de possibilitats a la sortida del requadre, per cada entrada (el valor que l'exploració ha assignat provisionalment a la casella prèvia), serà relativament baix, atès que la quantitat de valors 1... 9 compatibles en primera instància a cadascuna de les escases caselles lliures també serà baixa de mitjana. Si, a més, aquest requadre més poblat se situa al començament de l'exploració, la simplificació serà més eficient: tot i que el símil sigui un pèl groller, si volem eliminar la meitat de les fulles d'un arbre, acabarem abans tallant arran de tronc la meitat de les branques que arrencant, una per una, la meitat de les fulles.

En l'exemple de la falç i el martell, la disposició del SUDOKU-PROBLEMA ja respon a una bona ordenació (els requadres 9×3 tenen 10, 10 i 6 caselles plenes, de baix a dalt, i els requadres 3×9 en tenen 8, 9 i 9, d'esquerre a dreta, i com que el més ple dels primers en té més que el més ple dels segons, no caldrà transposar l'escaquer 9×9), en el sentit que, si anem emplenant d'esquerra a dreta i de baix a dalt, els valors admissibles apareixeran ràpidament en tots els passos sense que el dispositiu corrector hagi d'intervenir. Altra cosa seria que a l'usuari se li acudís començar omplint el requadre 3×9 superior: llavors el programa treballaria amb les caselles desplaçades al requadre 3×9 inferior, durant tota l'etapa. Per veure que l'últim plantejament és el correcte, suposem que després d'emplenar el requadre superior passa a la casella central (5,5), situació en què mesurarem el temps que triguen a aparèixer en el caseller 3×3 els 9 candidats vàlids (tots). Així podrem valorar la diferència de resultats en aquest cas, de primer suposant que **RE-MEMBER** actua sobre la disposició real de caselles (l'escaquer representat a la segona fila, a la dreta, amb els emplenaments que precedeixen el de la casella 5,5 (**8**) situats en el requadre 9×3 superior) i després suposant que el dispositiu optimitzador les desplaça cap avall (a la mateixa figura teniu una representació virtual d'aquests emplenaments, situada en el requadre 9×3 inferior): en el primer procés els nou candidats 1... 9 triguen 4 minuts i 10 segons a instal·lar-se en el caseller (de fet, el valor 1 apareix instantàniament, del 3 al 9 també, i és el 2 l'únic responsable de l'espera), mentre que el segon l'aparició conjunta és quasi instantània; hem triat la casella 5,5 com a patró de comparació perquè és exterior als requadres 9×3 superior i inferior, i així queda prou clar que la seva posició (la mateixa en dos casos amb resultats tan dispars) hi juga ben poc.

Però, com que una flor no fa estiu, aquest exemple podria ser l'excepció i no el paradigma, així que convindria improvisar una argumentació que, sense pretensions d'arribar a ser demostració inductiva rigorosa (que tampoc estaria a l'abast de l'autor), fos almenys una aproximació analògica a la comprensió del que pot passar a escala del sudoku, descrivint-ne el comportament a escala d'una o dues línies. Imaginem que només la primera línia d'un sudoku (la inferior) té caselles plenes (una d'elles serà la casella actual, amb el valor que estem examinant). Si hi ha **p** caselles plenes (**p ≤ 9**):

- La primera casella lliure admet **9 − p** valors.
- La segona " " " **8 − p** " .

.....

- La (8-p)-èsima casella lliure admet **2** valors.
- La (9-p)-èsima " " " **1** valor.

El nombre de combinacions a considerar (pel que fa a la línia) són **(9 − p)**! però, com que totes seran vàlides, ens plantaríem en la primera. Si hi tenim una casella plena, el nombre de combinacions serà **8! = 40.320**. Si el valor assignat fos un **5** i ocupés la casella 2ª, 5ª o 8ª (de forma semblant a abans, el lector primmirat pot suposar que aquestes caselles són les actuals i que **RE-MEMBER** està comprovant-hi si el **5** és un valor viable), la primera solució trobada en cada cas,

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 5 2   3 4 6   7 8 9      1 2 3   4 5 6   7 8 9      1 2 3   4 6 7   8 5 9
```
seria l'acceptada.

Si ara posem un **2** en la 2ª o 3ª línia, situant-lo com abans en la columna 2ª, 5ª o 8ª, el nombre de combinacions a considerar serà **6! × 8 × 7 × 6 = 241.920** (on **6! = 720** són les possibilitats d'omplir les 6 caselles no afectades pel **2**, i **8 × 7 × 6 = 336** les d'omplir les 3 afectades: variacions ordinàries amb 8 elements, d'ordre 3), la primera solució trobada en cada cas,

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 2 0   0 0 0   0 0 0      0 0 0   0 2 0   0 0 0      0 0 0   0 0 0   0 2 0
1 3 4   2 5 6   7 8 9      1 2 3   4 5 6   7 8 9      1 2 3   4 5 6   7 8 9
```
seria l'acceptada. Passa però que no totes les **9! − 241.920 = 120.960** possibilitats invalidades per la presència de la casella plena queden descartades a **MASCARA** per acció del filtre **(member N L)**, exterior a **RE-MEMBER**, sinó que haurà de ser aquesta funció la que en descobreixi la inviabilitat, amb l'aument exponencial consegüent del temps necessari per detectar-ho. Sortosament, en cap d'aquests tres casos no es dóna la circumstància, perquè l'ordenació de la llista **L1A9** (recordeu que tant **(member N L)** com **RE-MEMBER** estan dins de l'expressió **(foreach N L1A9 ...)**) i la dinàmica d'aquesta funció fan que abans es localitzin les solucions exposades. Però si, en comptes de ser un **2** fos un **8**, el fenòmen es manifestaria en el tercer requadre 3×3, per invalidació de la primera possibilitat assajada:

```
0 8 0   0 0 0   0 0 0      0 0 0   0 8 0   0 0 0      0 0 0   0 0 0   0 8 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 3   4 5 6   7 8 9      1 2 3   4 5 6   7 8 9      1 2 3   4 5 6   7 9 X
```

Com veieu, la primera solució trobada és acceptada en els dos primers casos, però en l'últim, la primera de les 241.920 solucions possibles és **1 2 3 4 5 8 6 7 9** perquè el primer emplenament endegat per **RE-MEMBER** es veurà frustrat en arribar a l'última casella, retornarà a la casella 8ª i 7ª però tampoc 1 2 3 4 5 6 9 7 **X** no haurà reeixit. En retornar a la 6ª i canviar de requadre, 1 2 3 4 5 7 6 9 **X** seguirà la mateixa sort, i no serà fins al quart intent que l'emplenament mostrat en negreta 5 línies enrera es podrà qualificar de solució, completant-se la línia. Precisament aquests retrocessos (8 caselles en l'exemple: 2 en el primer retrocés, 3 en el segon i 3 en el tercer) són el llast del qual convé desempallegar-se fins on sigui possible, per alleugerir el procés. És clar que, si ens quedéssim aquí, algú podria pensar que són els emplenaments els responsables del fenomen, i res més lluny de la realitat. Justament és el contrari: a mesura que anem emplenant un escaquer 9×9 inicialment buit es reduirà el nombre de retrocessos, i el que volem acabar d'aclarir és que aquesta reducció és més eficaç quan els emplenaments es practiquen en el requadre 9×3 inferior. Si havíem triat aquest fragment de sudoku era perquè ens permetia observar el fenomen sense passar de la 1ª línia, però ara en proposem un de sencer on es produeix en buit (repetirem la precisió d'abans: volem dir que, amb l'escaquer 9×9 encara buit, activarem la casella 1,1 i, ja en la cerca amb el valor **1**, es produirà el fenòmen).

Per tal que no trigui a aparèixer, us presentarem una SUDOKU-SOLUCIÓ molt semblant a l'anomenada SOLUCIÓ CANÒNICA (opció C del programa). I, perquè pogueu comparar-les millor (tenen idèntic el requadre 9×3 inferior), posarem la CANÒNICA en primer lloc, després la solució que ens convé utilitzar ara, que anomenarem NATURMÍN, i, *per torna* (per completar un espai on n'hi caben tres, aclariment que els nascuts abans del 1960, com l'autor, no necessiten), una cosina que anomenarem NATURMAX. Ras i curt, definirem NATURMÍN com la solució que obtindríem emplenant l'escaquer des de la casella 1,1, avançant d'esquerra a dreta i de baix a dalt, i assignant a totes les caselles el més petit dels valors proposats per **MASCARA** (caseller 3×3). La particularitat d'aquest sudoku (que justifica que el qualifiquem de "NATURal") és que l'emplenament virtual amb què **RE-MEMBER** confirmarà la compatibilitat del valor de qualsevol casella (en l'exploració que li correspongui com a candidat) serà idèntic a la SUDOKU-SOLUCIÓ, amb independència que l'activació de la casella tingui lloc en un escaquer buit o parcialment emplenat amb valors de la solució. Com era de preveure, NATURMAX és la solució que obtindríem seguint el mateix ordre d'emplenament però fent-ho amb el valor més gran. Per aconseguir en aquest cas que l'emplenament virtual amb què **RE-MEMBER** confirmarà la compatibilitat del valor de qualsevol casella sigui idèntic a SUDOKU-SOLUCIÓ, n'hi haurà prou a substituir en la funció **(foreach E L1A9 ...)** per **(foreach E (reverse L1A9) ...)**. Aquí les teniu:

9 1 2 3 4 5 6 7 8	**9** 7 8 5 **3** 1 6 4 **2**	1 3 2 5 7 9 4 6 8
6 7 8 9 1 2 3 4 5	6 **4** 2 **9** 7 **8** 5 **3** 1	4 6 8 1 3 2 5 7 9
3 4 5 6 7 8 9 1 2	**5** 3 1 6 **4** 2 9 7 **8**	5 7 9 4 6 8 1 3 2
8 9 1 2 3 4 5 6 7	**8** 9 7 2 **1** 4 3 6 **5**	2 1 3 8 9 6 7 4 5
5 6 7 8 9 1 2 3 4	3 **6** 5 **8** 9 **7** 2 **1** 4	7 4 5 2 1 3 8 9 6
2 3 4 5 6 7 8 9 1	**2** 1 4 3 **6** 5 8 9 **7**	8 9 6 7 4 5 2 1 3
7 8 9 1 2 3 4 5 6	**7** 8 9 1 **2** 3 4 5 **6**	3 2 1 9 8 7 6 5 4
4 5 6 7 8 9 1 2 3	4 **5** 6 **7** 8 **9** 1 **2** 3	6 5 4 3 2 1 9 8 7
1 2 3 4 5 6 7 8 9	**1** 2 3 4 **5** 6 7 8 **9**	9 8 7 6 5 4 3 2 1
Solució CANÒNICA	Solució NATURMÍN	Solució NATURMAX

Farem un parèntesi per destacar algunes curiositats:
- El requadre 9×3 inferior de NATURMAX és simètric del de NATURMÍN (i, pel que hem dit abans, del de CANÒNICA).
- Les caselles homòlogues de NATURMÍN i NATURMAX sumen 10 (en el primer capítol, *Inventariar les solucions: un cul-de-sac*, les anomenàvem *complementàries a 10*), cosa absolutament previsible perquè comparant les primeres línies es veu que les dues solucions estan relacionades per una transformació de tipus PV (Permutació entre Valors 1 a 9) on $\mathbf{V_{max} = 10 - V_{min}}$.
- En cada requadre 9×3 es repeteixen 3 tercets: a CANÒNICA això s'esdevenia per construcció, però a NATURMÍN i a NATURMAX no.
- A NATURMÍN el tercet format per **7**, **8** i **9** apareix (permutat circularment de forma diferent en cada requadre 9×3) a tots els requadres 3×3, i a NATURMAX passa el mateix amb els valors **1**, **2** i **3** (cosa que es dedueix del penúltim enunciat).

Tornant al nostra tema i a la hipòtesi de l'escaquer buit, que abordarem activant la casella 1,1, on **RE-MEMBER** començarà comprovant-hi el valor **1** (que per això està en negreta), suposarem que la 1ª fila ja està completa i que la funció ha emplenat virtualment el primer tercet de la 2ª amb els valors més baixos amb què podia fer-ho (4, 5 i 6) i també el segon, amb el mateix supòsit (1, 2 i 3): en passar a la casella 7,2, **RE-MEMBER** es troba que no hi ha cap valor compatible (esquerra) i ha de recular fins la casella 6,2, on hi prova el 7, que ve després del 3 com a valor compatible en primera instància, però es torna a quedar en blanc en 7,2 (centre); després d'assajar en 6,2 amb els valors 8 i 9 i amb idèntics resultats, recula una posició més, i en 5,2 prova els valors 3, 7, 8 i 9 (dreta) però sempre, en arribar a la casella 7,2, ensopega amb el mateix roc; finalment recula fins 4,2 i, després d'haver-hi assajat els valors 2 i 3 infructuosament, hi prova el 7, el 8 en 5,3 i el 9 en 6,3. Ara sí que pot completar la 2ª línia encadenant-hi els valors 1, 2 i 3 (veieu-la a la Solució NATURMÍN).

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
4 5 6 1 2 3 **X** 0 0	4 5 6 1 2 7 **X** 0 0	4 5 6 1 7 8 **X** 0 0
1 2 3 4 5 6 7 8 9	**1** 2 3 4 5 6 7 8 9	**1** 2 3 4 5 6 7 8 9

Deixem al lector puntimirat el recompte de les posicions que **RE-MEMBER** ha hagut de retrocedir abans de deixar explorades satisfactòriament les dues primeres línies, però amb això haurà quedat palès que les ziga-zagues a la trajectòria d'exploració no són privatives de les situacions amb caselles ocupades.

De NATURMÍN encara en podem treure més suc i, per si el lector encara no s'ensuma quina en portem de cap, aclarim que ens interessaria saber quants passos enrera ha de donar **RE-MEMBER** per poder oferir el **1** com a valor assignable a la casella 1,1. Naturalment, n'exclourem la seqüencia ininterrompuda de passos enrera de l'última a la primera casella lliure, immediatament després d'haver-les emplenat totes per primera vegada (haver baixat tots els nivells de recursió, des de la primera fulla fins l'arrel, descens desfermat per l'activació de **(COMPLET-9*9 R-9*9)** i mantingut en ràpel pel de la variable **OK**). Per començar, no només ens limitarem al trajecte equivalent a NATURMÍN (valor **1** en 1,1, amb prioritat per als candidats més petits) i, ja de passada a l'equivalent a NATURMAX (valor **9** en 1,1, amb prioritat per als candidats més grans), sinó que en tots dos casos imaginarem trajectòries de cerca corresponents als vuit valors restants, també assignats a la casella 1,1 (parlem d'imaginar perquè, a diferència de NATURMÍN i NATURMAX, representats a la pàgina precedent, aquests resultats no s'han arribat a plasmar). Els retrocessos en cada un dels tres requadres 9×3 s'exposen com a termes d'una suma, seguides del total

nombre_retorns_req_1 + nombre_retorns_req_2 + nombre_retorns_req_3 = total

i les de NATURMÍN (**1**) i NATURMAX (**9**) es destaquen en negreta, cosa que ajudarà a copsar la simetria en nombre de retrocessos entre les dues taules (circumstància

previsible si pensem que no només sumen 10 les caselles homòlogues de NATURMÍN i NATURMAX sinó que, en cada una i per a cada valor, **RE-MEMBER** visita els candidats en ordre invers -de 1 a 9 en NATURMÍN i de 9 a 1 en NATURMAX- i les trajectòries virtuals seran idèntiques quant a traçat, però amb valors homòlegs que sumen 10):

Exploració casella 1,1 amb l'escaquer buit, seguint patró habitual (NATURMÍN)

```
- Provant-hi el candidat 1 hi haurà   246 +    49 +    15 =   310 retrocessos
- Provant-hi el candidat 2 hi haurà   246 +    49 +    15 =   310 retrocessos
- Provant-hi el candidat 3 hi haurà   246 +    61 +    15 =   322 retrocessos
- Provant-hi el candidat 4 hi haurà   246 +    62 +    87 =   395 retrocessos
- Provant-hi el candidat 5 hi haurà   246 +    36 +    95 =   377 retrocessos
- Provant-hi el candidat 6 hi haurà   246 +    24 +     5 =   275 retrocessos
- Provant-hi el candidat 7 hi haurà   184 +    37 +    95 =   316 retrocessos
- Provant-hi el candidat 8 hi haurà   169 +    65 +    95 =   329 retrocessos
- Provant-hi el candidat 9 hi haurà   164 +    64 +    95 =   323 retrocessos
-------------------------------------------------------------------------
  Sumant retrocessos per requadres: 1.993 +   447 +   517 = 2.957 retrocessos
```

Exploració de la casella 1,1 amb l'escaquer buit, seguint el patró NATURMAX

```
- Provant-hi el candidat 1 hi haurà   164 +    64 +    95 =   323 retrocessos
- Provant-hi el candidat 1 hi haurà   169 +    65 +    95 =   329 retrocessos
- Provant-hi el candidat 1 hi haurà   184 +    37 +    95 =   316 retrocessos
- Provant-hi el candidat 1 hi haurà   246 +    24 +     5 =   275 retrocessos
- Provant-hi el candidat 1 hi haurà   246 +    36 +    95 =   377 retrocessos
- Provant-hi el candidat 1 hi haurà   246 +    62 +    87 =   395 retrocessos
- Provant-hi el candidat 1 hi haurà   246 +    61 +    15 =   322 retrocessos
- Provant-hi el candidat 1 hi haurà   246 +    49 +    15 =   310 retrocessos
- Provant-hi el candidat 1 hi haurà   246 +    49 +    15 =   310 retrocessos
-------------------------------------------------------------------------
  Sumant retrocessos per requadres: 1.993 +   447 +   517 = 2.957 retrocessos
```

Adaptar el programa per aconseguir aquestes dades ha estat molt senzill: n'hi ha hagut prou a incorporar-hi la funció **INFORMA**,

```
(defun INFORMA (TOT NO)
  (if (not TOT) (setq *TOT-1* (+ *TOT-1* *REQ-1*)
                      *TOT-2* (+ *TOT-2* *REQ-2*)
                      *TOT-3* (+ *TOT-3* *REQ-3*)))
  (prompt (strcat "\n" (if TOT "T" (itoa N)) (if NO "<" ">")
                  " "   (itoa (if TOT *TOT-1* *REQ-1*))
                  " + " (itoa (if TOT *TOT-2* *REQ-2*))
                  " + " (itoa (if TOT *TOT-3* *REQ-3*))
                  " = " (itoa (if TOT (+ *TOT-1* *TOT-2* *TOT-3*)
                                      (+ *REQ-1* *REQ-2* *REQ-3*)))))
  (if (not TOT) (setq  *REQ-1* 0 *REQ-2* 0 *REQ-3* 0)))
```

intervenir **MASCARA**, de la qual laurem eliminat la primera expressió **(if POST ...)** i canviat el contingut de l'última, inserint-hi les expressions subratllades,

```
(defun MASCARA (L / INI OK JJ 9**9 0-**9*9** 0-LLL 0-L)
  (setq LL () SS (ssadd))
  (setq *REQ-1* 0 *REQ-2* 0 *REQ-3* 0
        *TOT-1* 0 *TOT-2* 0 *TOT-3* 0)
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER **9*9** LLL)))
        (progn
          (INFORMA () ())
          (if POST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         .......................
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
```

```
        (progn
          (if POST (TREU-RETOL))))
          (if (member N L) (INFORMA () T)))
  (if POST (INFORMA T ())))
```

i fer el mateix a **RE-MEMBER** (que, en ser més curta, reproduïm sencera)

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI () *NIV* 1))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)) *NIV* (1+ *NIV*))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R))
                    (if (not OK)
                        (progn
                          (cond ((< *NIV* 28) (setq *REQ-1* (1+ *REQ-1*)))
                                ((< *NIV* 55) (setq *REQ-2* (1+ *REQ-2*)))
                                (     T       (setq *REQ-3* (1+ *REQ-3*))))
                          (setq *NIV* (1- *NIV*)))))))))))
  OK)
```

Els resultats d'abans ja comencen a ser significatius, perquè encara que NATURMÍN i NATURMAX siguin dos sudokus molt especials (dintre d'un ordre d'emplenament ja establert, la tria entre els valors possibles no és arbitrària sinó que respon a una llei igualment predeterminada), en un cert sentit representen dues tendències oposades, entre les quals se situarien tots els demés (unes vegades el valor de la casella actual coincidirà amb el mínim dels compatibles amb les caselles ocupades, unes altres amb el màxim i en la resta d'ocasions se situarà entre ambdós valors). A més, convé recordar que aquests valors no representen el nombre de ziga-zagues (de diferent amplitud) sinó el de salts d'una casella lliure a la casella lliure precedent, de manera que es poden interpretar com una aproximació a la mesura del temps invertit en el recorregut. I, si alguna conclusió destaca de la lectura, és que en buit (és a dir, en començar el procés d'emplenament), el requadre 9×3 més castigat és l'inferior; pel que fa al central i al superior encara no ens hi volem pronunciar, malgrat tenir l'últim alguns retrocessos més que el precedent, atès que la diferència entre els d'aquests requadres (el segon i tercer) és d'un ordre inferior a les diferències respecte als del primer. I no parlem pas de les dades desglossades per valors d'assignació, que potser més endavant ens seran útils per destacar alguna qüestió de detall, sinó de les globals, situades al final de cada taula: si volem comparar els temps d'espera en diferents caselles no només hem de comptar el corresponent al valor que triarem sinó el degut al conjunt de candidats (valors compatibles en primera instància). Per això, d'ara endavant ens limitarem a reproduir el total de retrocessos associats a l'emplenament d'una casella (això sí: desglossats segons el requadre 9×3 on es produeixin), i calcularem manualment i hi inclourem la suma d'aquests totals. Només en algun cas singular (com el dels retrocessos a 4,2 (**1**), 5,2 (**2**) i 6,2 (**3**), descrits quan analitzàvem l'emplenament de les dues primeres files de NATURMÍN), anirem al detall dels retrocessos deguts a cada candidat, fins i tot destacant els que han acabat refusats per **RE-MEMBER**: si repasseu amb deteniment **INFORMA** i l'adaptació de **MASCARA**, veureu que aquests proscrits es distingeixen dels compatibles i del total (7**>**, 8**>**, 9**>** o T**>**) en l'ús d'un altre caràcter separador (1**<**, 2**<** o 3**<**). Però això serà al final del capítol.

A partir d'aquí, podríem conjecturar que s'escau emplenar abans el primer requadre (ho faci l'usuari o s'espavili el programa perquè sempre sigui així, que és cap on ens encaminem), precisament per contrarestar la incidència més negativa que sembla tenir en el comportament conjunt. Però, sense pretendre passar de la conjectura a la demostració, sí que podem donar-li més consistència assajant altres posicions i considerant emplenaments múltiples, perquè gairebé el mateix algorisme pot servir, no ja per copsar les diferències de temps que es registren concentrant l'ocupació en un o altre requadre 9×3 (que això prou que ho notarem), sinó per entreveure'n

les causes. És clar que ara ja no podrem refiar-nos de saber la posició, comptant casella a casella avanços i retrocessos, als efectes d'adscriure'ls a un o altre requadre 9×3. Per no incloure en el recompte les caselles ocupades que trobem pel camí hauríem de complicar-nos la vida intervenint **COMPLET** o **ACTUALITZA**, i surt més a compte capturar de **R-P** la coordenada Y de les caselles retrocedides:

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R))
                    (if (not OK)
                        (progn
                          (setq *NIV* (cadr R-P))
                          (cond ((< *NIV* 3) (setq *REQ-1* (1+ *REQ-1*)))
                                ((< *NIV* 6) (setq *REQ-2* (1+ *REQ-2*)))
                                (    T     (setq *REQ-3* (1+ *REQ-3*)))))))))))))
  OK)
```

Per estrenar la nova eina (tot i que en aquest cas encara hauríem pogut aprofitar l'antiga), començarem activant en buit 30 caselles aïllades (una cada vegada) que ocuparan les mateixes 10 posicions relatives en el primer, segon i tercer requadre 9×3 (la 1ª, 5ª, 9ª, 11ª, 13, 15ª, 17ª, 19ª, 23ª i 27ª del requadre, representades en negreta en la Solució NATURMÍN, a la pàgina 175). Les 10 caselles explorades en cada requadre, cadascuna representada per la tríada de retrocessos esdevinguts en l'exploració, classificats per requadres tal i com s'ha convingut, les presentarem aplegades en tres taules. Al lector no li caldrà commutar a finestra de text per comprovar si les quatre dades associades a una casella i que figuren en les taules (3 sumes parcials i el total) coincideixen amb les generades pel programa adaptat, perquè aquesta informació surt en l'última línia i serà visible des de la pantalla gràfica, tot i que és lliure de fer-ho si l'interessa conèixer quants retrocessos s'han produït explorant la viabilitat de cada valor 1... 9 candidat. Ara bé, si ho fa per comprovar si els tres termes i la suma corresponentss al valor d'assignació (el que figura a NATURMÍN) sempre són els mateixos, s'ho ben pot estalviar, perquè una cosa és afirmar, com quan presentàvem NATURMÍN, que l'emplenament virtual amb què **RE-MEMBER** determina la viabilitat d'un valor coincideix amb aquesta solució, tant si la casella s'activa amb l'escaquer buit com si aquest ja està mig emplenat amb valors de NATURMÍN, i una altra pretendre que la trajectòria de l'emplenament sempre tindrà la mateixa longitud (ziga-zagues incloses), mesurada en retrocessos: fins i tot amb l'escaquer buit, com ara veurem, no té per què fer igual de llarg l'itinerari de verificació de compatibilitat del candidat **1** a la casella 1,1 (amb 246 + 49 + 15 = 310 retrocessos) que la d'una posició més avançada (fila, columna i requadre 3×3 diferents) com és la del valor **2** a la casella 5,3 (130 + 49 + 15 = = 194 retrocessos), per no moure'ns d'un mateix requadre 9×3 i poder referir-nos a uns fets coneguts. En el primer cas, recordeu quan teníem virtualment ocupades les 12 primeres caselles i emplenàvem el tercet 4,2, 5,2 i 6,2 amb els valors **1-2-3**: la casella 7,2 no admetia **7**, **8** ni **9** (**1-2-3-X**-0-0), i anàvem assajant sense èxit els sextets **1-2-7-3-X**-0, **1-2-7-X**-0-0, **1-2-8-3-X**-0, **1-2-8-X**-0-0, **1-2-9-3-X**-0, **1-2-9-X**-0-0, **1-3-2-X**-0-0, **1-3-7-2-X**-0, **1-3-7-X**-0-0, **1-3-8-2-X**-0, **1-3-8-X**-0-0, **1-3-9-2-X**-0, **1-3-9-X**-0-0, **1-7-2-3-X**-0, **1-7-2-X**-0-0, **1-7-3-2-X**-0, **1-7-3-X**-0-0, **1-8-2-3-X**-0, **1-8-2-X**-0-0, **1-8-3-2-X**-0, **1-8-3-X**-0-0, **1-8-7-2-3-X**, **1-8-7-2-X**-0, **1-8-7-3-2-X**, **1-8-7-3-X**-0, **1-8-7-X**-0-0, **1-8-9-2-3-X**, **1-8-9-2-X**-0, **1-8-9-3-2-X**, **1-8-9-3-X**-0, **1-8-9-X**-0-0, **1-9-2-3-X**-0, **1-9-2-X**-0-0, **1-9-3-2-X**-0, **1-9-3-X**-0-0, **1-9-7-2-3-X**, **1-9-7-2-X**-0, **1-9-7-3-2-X**, **1-9-7-3-X**-0, **1-9-8-2-3-X**, **1-9-8-2-X**-0, **1-9-8-3-2-X**, **1-9-8-3-X**-0, **1-9-8-X**-0-0, **2-1-3-X**-0-0, **2-1-7-3-X**-0, **2-1-7-X**-0-0, **2-1-8-3-X**-0, **2-1-8-X**-0-0, **2-1-9-3-X**-0, **2-1-9-X**-0-0, **2-3-1-X**-0-0, **2-3-7-1-X**-0, **2-3-7-X**-0-0, **2-3-8-1-X**-0, **2-3-8-X**-0-0, **2-3-9-1-X**-0, **2-3-9-X**-0-0, **2-7-1-3-X**-0, **2-7-1-X**-0-0, **2-7-3-1-X**-0, **2-7-3-X**-0-0, **2-8-1-3-X**-0, **2-8-1-X**-0-0, **2-8-3-1-X**-0, **2-8-3-X**-0-0, **2-8-7-1-3-X**, **2-8-7-1-X**-0, **2-8-7-3-1-X**, **2-8-7-3-X**-0, **2-8-7-X**-0-0,

2-8-9-1-3-X, **2-8-9-1-X**-0, **2-8-9-3-1-X**, **2-8-9-3-X**-0, **2-8-9-X**-0-0, **2-9-1-3-X**-0, **2-9-1-X**-0-0, **2-9-3-1-X**-0, **2-9-3-X**-0-0, **2-9-7-1-3-X**, **2-9-7-1-X**-0, **2-9-7-3-1-X**, **2-9-7-3-X**-0, **2-9-8-1-3-X**, **2-9-8-1-X**-0, **2-9-8-3-1-X**, **2-9-8-3-X**-0, **2-9-8-X**-0-0, **3-1-2-X**-0-0, **3-1-7-2-X**-0, **3-1-7-X**-0-0, **3-1-8-2-X**-0, **3-1-8-X**-0-0, **3-1-9-2-X**-0, **3-1-9-X**-0-0, **3-2-1-X**-0-0, **3-2-7-1-X**-0, **3-2-7-X**-0-0, **3-2-8-1-X**-0, **3-2-8-X**-0-0, **3-2-9-1-X**-0, **3-2-9-X**-0-0, **3-7-1-2-X**-0, **3-7-1-X**-0-0, **3-7-2-1-X**-0, **3-7-2-X**-0-0, **3-8-1-2-X**-0, **3-8-1-X**-0-0, **3-8-2-1-X**-0, **3-8-2-X**-0-0, **3-8-7-1-2-X**, **3-8-7-1-X**-0, **3-8-7-2-1-X**, **3-8-7-2-X**-0, **3-8-7-X**-0-0, **3-8-9-1-2-X**, **3-8-9-1-X**-0, **3-8-9-2-1-X**, **3-8-9-2-X**-0, **3-8-9-X**-0-0, **3-9-1-2-X**-0, **3-9-1-X**-0-0, **3-9-2-1-X**-0, **3-9-2-X**-0-0, **3-9-7-1-2-X**, **3-9-7-2-X**-0, **3-9-7-2-1-X**, **3-9-7-2-X**-0, **3-9-8-1-2-X**, **3-9-8-1-X**-0, **3-9-8-2-1-X**, **3-9-8-2-X**-0, **3-9-8-X**-0-0, **7-1-2-3-X**-0, **7-1-2-X**-0-0, **7-1-3-2-X**-0, **7-1-3-X**-0-0, **7-1-8-2-3-X**, **7-1-8-2-X**-0, **7-1-8-3-2**-0, **7-1-9-2-3-X**, **7-1-9-2-X**-0, **7-1-9-3-2-X**, **7-1-9-3-X**-0, **7-2-1-3-X**-0, **7-2-1-X**-0-0, **7-2-3-1-X**-0, **7-2-3-X**-0-0, **7-2-8-1-3-X**, **7-2-8-1-X**-0, **7-2-8-3-1-X**, **7-2-8-3-X**-0, **7-2-9-1-3-X**, **7-2-9-1-X**-0, **7-2-9-3-1-X**, **7-2-9-3-X**-0, **7-3-1-2-X**-0, **7-3-1-X**-0-0, **7-3-2-1-X**-0, **7-3-2-X**-0-0, **7-3-8-1-2-X**, **7-3-8-1-X**-0, **7-3-8-2-1-X**, **7-3-8-X**-0-0, **7-3-9-1-2-X**, **7-3-9-1-X**-0, **7-3-9-2-1-X** i **7-3-9-2-X**-0, fins a poder completar sextet i línia amb **7-8-9-1-2-3**. En total són 170 intents fallits, la majoria dels quals comporten un retrocés per anar al següent, però alguns passos en comporten dos o tres. Doncs bé: si observem que d'aquests no menys de 170 retrocessos n'hi ha no menys de 89 que condueixen a assignar un **2** a la caselles 4,2, 5,2 o 6,2, retrocessos que ja no es produiran en el segon cas, perquè la presència virtual del candidat **2** a la casella 5,3 (que és en el mateix requadre 3×3) ho impedirà. I és que estrictament no hauríem de parlar d'activar *en buit* una casella, ja que allò que fa **RE-MEMBER** és justament estudiar la viabilitat d'un valor 1... 9 en una casella, partint de la hipòtesi que aquesta assignació s'ha produït. Per aquesta raó, en comptes d'escriure, com abans,

- Provant-hi el candidat...
- Provant-hi el candidat...
- Provant-hi el candidat...

.....

hi posarem obertament

- Si només omplim 1,1 (**1**)...
- Si només omplim 5,1 (**5**)...
- Si només omplim 9,1 (**9**)...

.....

expressió que més endavant canviarem per

- Si només omplim 1,1 (**1**)...
- Si també omplim 5,1 (**5**)...
- Si també omplim 9,1 (**9**)...

.....

per indicar que cada casella emplenada s'afegeix a les anteriors. De moment, però, veiem el resultat d'emplenar-les separadament:

Exploració caselles 1r requadre amb l'escaquer buit, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà **1.993** + **447** + **517** = **2.957** retrocessos
- Si només omplim 5,1 (**5**) hi haurà **1.584** + **218** + **515** = **2.317** retrocessos
- Si només omplim 9,1 (**9**) hi haurà **954** + **598** + **291** = **1.843** retrocessos
- Si només omplim 2,2 (**5**) hi haurà **1.584** + **795** + **477** = **2.856** retrocessos
- Si només omplim 4,2 (**7**) hi haurà **654** + **608** + **214** = **1.476** retrocessos
- Si només omplim 6,2 (**9**) hi haurà **654** + **471** + **157** = **1.282** retrocessos
- Si només omplim 8,2 (**2**) hi haurà **312** + **1.072** + **180** = **1.564** retrocessos
- Si només omplim 1,3 (**7**) hi haurà **954** + **1.104** + **302** = **2.360** retrocessos
- Si només omplim 5,3 (**2**) hi haurà **1.545** + **742** + **232** = **2.519** retrocessos
- Si només omplim 9,3 (**6**) hi haurà **3.894** + **690** + **201** = **4.785** retrocessos

---

Sumant retrocessos per requadr: **14.128** + **6.795** + **3.086** = **23.959** retrocessos

Exploració caselles 2n requadre amb l'escaquer buit, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà **2.214** + **489** + **953** = **3.656** retrocessos
- Si només omplim 5,4 (**6**) hi haurà **1.986** + **332** + **176** = **2.494** retrocessos
- Si només omplim 9,4 (**7**) hi haurà **2.217** + **271** + **198** = **2.686** retrocessos
- Si només omplim 2,5 (**6**) hi haurà **2.214** + **452** + **267** = **2.933** retrocessos
- Si només omplim 4,5 (**8**) hi haurà **2.028** + **350** + **139** = **2.517** retrocessos
- Si només omplim 6,5 (**7**) hi haurà **1.797** + **224** + **234** = **2.255** retrocessos
- Si només omplim 8,5 (**1**) hi haurà **2.160** + **232** + **283** = **2.675** retrocessos
- Si només omplim 1,6 (**8**) hi haurà **2.214** + **358** + **253** = **2.825** retrocessos

- Si només omplim 5,6 (**1**) hi haurà **1.986** + **734** + **284** = **3.004** retrocessos
- Si només omplim 9,6 (**5**) hi haurà **2.217** + **1.013** + **204** = **3.434** retrocessos

---

Sumant retrocessos per requadr: **21.033** + **4.455** + **2.991** = **28.479** retrocessos

Exploració caselles 3r requadre amb l'escaquer buit, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà **2.214** + **489** + **635** = **3.338** retrocessos
- Si només omplim 5,7 (**4**) hi haurà **1.986** + **363** + **173** = **2.522** retrocessos
- Si només omplim 9,7 (**8**) hi haurà **2.217** + **1.009** + **140** = **3.366** retrocessos
- Si només omplim 2,8 (**4**) hi haurà **2.214** + **476** + **163** = **2.853** retrocessos
- Si només omplim 4,8 (**9**) hi haurà **2.028** + **763** + **164** = **2.955** retrocessos
- Si només omplim 6,8 (**8**) hi haurà **1.797** + **399** + **118** = **2.314** retrocessos
- Si només omplim 8,8 (**3**) hi haurà **2.160** + **438** + **97** = **2.695** retrocessos
- Si només omplim 1,9 (**9**) hi haurà **2.214** + **489** + **343** = **3.046** retrocessos
- Si només omplim 5,9 (**3**) hi haurà **1.986** + **363** + **218** = **2.567** retrocessos
- Si només omplim 9,9 (**2**) hi haurà **2.217** + **1.009** + **466** = **3.692** retrocessos

---

Sumant retrocessos per requadr: **21.033** + **5.798** + **2.517** = **29.348** retrocessos

Tot i que les diferències no són exagerades, queda clar que, de mitjana, confirmar la validesa d'un candidat a una casella comporta més retrocessos (temps de cerca) considerant-la situada en el segon requadre 9×3 que en el primer, i en comporta més considerant-la situada en el tercer requadre que en el segon. ¿Això vol dir que abans, quan permutàvem requadres 9×3 de manera que la casella actual sempre fos en el terç inferior de l'escaquer, no anàvem tan errats? Doncs més aviat no, perquè encara que això respongui a una lògica en el moment inicial, amb l'escaquer encara buit, veurem que a mesura que l'anem omplint va perdent sentit.

Ara passem a omplir les mateixes 10 caselles de cada requadre, però aplegant-les. Si no és que sentiu curiositat per algun valor parcial d'assignació, estalvieu-vos la commutació a pantalla de text per veure el llistat complet, després d'activar cada casella, perquè només ens interessa l'última línia (la de totals) que surt en pantalla gràfica. Com abans, aquests són els valors que figuren a les tres taules:

Exploració i emplenament de 10 caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà **1.993** + **447** + **517** = **2.957** retrocessos
- Si també omplim 5,1 (**5**) hi haurà **1.338** + **207** + **386** = **1.931** retrocessos
- Si també omplim 9,1 (**9**) hi haurà **882** + **409** + **152** = **1.443** retrocessos
- Si també omplim 2,2 (**5**) hi haurà **1.338** + **733** + **388** = **2.459** retrocessos
- Si també omplim 4,2 (**7**) hi haurà **594** + **572** + **200** = **1.366** retrocessos
- Si també omplim 6,2 (**9**) hi haurà **135** + **400** + **143** = **678** retrocessos
- Si també omplim 8,2 (**2**) hi haurà **14** + **319** + **140** = **437** retrocessos
- Si també omplim 1,3 (**7**) hi haurà **12** + **399** + **135** = **522** retrocessos
- Si també omplim 5,3 (**2**) hi haurà **37** + **439** + **144** = **510** retrocessos
- Si també omplim 9,3 (**6**) hi haurà **21** + **76** + **42** = **620** retrocessos

---

Sumant retrocessos per requadre: **6.356** + **4.287** + **2.316** = **12.959** retrocessos

Exploració i emplenament de 10 caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà **2.214** + **489** + **953** = **3.656** retrocessos
- Si també omplim 5,4 (**6**) hi haurà **1.171** + **309** + **159** = **2.239** retrocessos
- Si també omplim 9,4 (**7**) hi haurà **1.724** + **223** + **122** = **2.069** retrocessos
- Si també omplim 2,5 (**6**) hi haurà **1.968** + **300** + **187** = **2.455** retrocessos
- Si també omplim 4,5 (**8**) hi haurà **1.782** + **338** + **184** = **2.304** retrocessos
- Si també omplim 6,5 (**7**) hi haurà **1.571** + **131** + **183** = **1.885** retrocessos
- Si també omplim 8,5 (**1**) hi haurà **1.182** + **43** + **168** = **1.393** retrocessos
- Si també omplim 1,6 (**8**) hi haurà **1.330** + **57** + **168** = **1.555** retrocessos
- Si també omplim 5,6 (**1**) hi haurà **882** + **115** + **124** = **1.121** retrocessos
- Si també omplim 9,6 (**5**) hi haurà **998** + **1.165** + **101** = **2.264** retrocessos

---

Sumant retrocessos per requadr: **15.422** + **3.170** + **2.349** = **20.941** retrocessos

La conclusió més immediata, si comparem aquestes dues taules amb les homòlogues de la sèrie precedent, és que es confirmen les tesis que havíem avançat: a mesura que emplenem un escaquer 9×9 inicialment buit, es reduirà el nombre de retrocessos;

aquesta reducció és més eficaç quan els emplenaments es practiquen en el requadre 9×3 inferior (12.959 / 23.959 = 0,54) que en el mitjà (20.941 / 28.479 = 0,74). Ens agradaria que la taula corresponent al requadre superior reblés aquest clau, però dissortadament no serà així. Abans de comprovar-ho, però, caldria matisar la valoració que hem fet de les dades obtingudes dels dos primers requadres, perquè, igual que hem dit que la reducció deguda a l'emplenament del requadre és major en el primer, podíem haver dit que la diferència entre els costos per confirmar un candidat a una casella del primer o del segon requadre 9×3 encara s'ha accentuat (de 23.959 a 28.479, que representa un augment del 19%, s'ha passat de 12.959 a 20.941, que en representa un del 62%), constatació que reforçaria la conveniència de desplaçar sempre la casella actual cap al primer requadre. Per adonar-se que la ubicació de la casella actual va perdent importància davant de la del centre de gravetat de les caselles emplenades prèviament, farem dues proves ràpides (en el sentit de limitar-nos a 2 caselles per prova, per no farcir encara més el text amb taules de 10), una amb el primer requadre 9×3 amb les 10 caselles ja emplenades i l'altra amb el segon requadre emplenat anàlogament): després d'haver ocupat les 10 caselles del primer requadre,

- si activem 5,5 (centre 2n requadre) hi haurà **36** + **173** + **153** = **362** retrocessos
- si activem 5,8 (centre 3r requadre) hi haurà **36** + **276** + **127** = **439** retrocessos

i, després d'haver ocupat les 10 caselles del segon requadre,

- si activem 5,2 (centre 1r requadre) hi haurà **386** + **194** + **136** = **716** retrocessos
- si activem 5,8 (centre 3r requadre) hi haurà **869** + **39** + **109** = **1.107** retrocessos

Les oscil·lacions dintre de cada parell de valors (del 19% i 33%, respectivament) són irrellevants al costat de l'existent entre els seus valors mitjans (del 78%), estimació que ens fa concloure que la posició del centre geomètric de les caselles plenes condicionarà força més la durada de la cerca **RE-MEMBER** a la casella actual que la posició d'aquesta casella. Feta aquesta precisió, anem al tercer requadre:

Exploració i emplenament de 10 caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**2.214** +	**489** +	**635** =	**3.338** ret.
- Si també omplim 5,7 (**4**) hi haurà	**1.740** +	**338** +	**137** =	**2.215** ret.
- Si també omplim 9,7 (**8**) hi haurà	**1.725** +	**451** +	**92** =	**2.268** ret.
- Si també omplim 2,8 (**4**) hi haurà	**1.968** +	**407** +	**138** =	**2.513** ret.
- Si també omplim 4,8 (**9**) hi haurà	**1.782** +	**586** +	**122** =	**2.044** ret.
- Si també omplim 6,8 (**8**) hi haurà	**1.346** +	**307** +	**155** =	**1.808** ret.
- Si també omplim 8,8 (**3**) hi haurà	**1.098** +	**957** +	**1.460** =	**3.515** ret.
- Si també omplim 1,9 (**9**) hi haurà	**1.239** +	**995.732** +	**839.620** =	**1.836.591** ret.
- Si també omplim 5,9 (**3**) hi haurà	**970** +	**27.174** +	**20.996** =	**49.140** ret.
- Si també omplim 9,9 (**2**) hi haurà	**?** +	**?** +	**?** =	**?** ret.
Sumant retrocessos per requadre:	**?** +	**?** +	**?** =	**?** ret.

Consignarem que el sobtat increment que es produeix a 1,9 és degut al candidat 6 i, en menor mesura, a 3:

```
3> 177 +   8.637 +   8.256 =    17.070
6> 177 + 895.544 + 829.969 = 1.815.690
```

I, com probablement haureu imaginat, ens hem quedat amb les ganes de conèixer els retrocessos produïts en activar 9,9, perquè al cap de 6 hores l'autor va desistir. Sí que es pot veure (amb una segona passada on l'expressió **(foreach N L1A9 ...)** de **MASCARA** s'havia canviat per **(foreach N (reverse L1A9) ...)**) que el responsable és el **5**, reproduint aquí el desglossament per candidats que surt en pantalla de text:

```
1> 155 +      35 +      9 =     199
2> 155 +      35 +      0 =     190
4> 155 +      36 +      9 =     200
5>   ? +       ? +      ? =       ?
6> 156 +     500 +    280 =     936
7> 155 + 127.041 + 36.551 = 163.747
T>   ? +       ? +      ? =       ?
```

Introduint a **RE-MEMBER** dispositius similars als que ens proporcionen les dades amb què estem jugant, i centrant-nos en la verificació dels candidats **3** i **6** per a la casella 1,9 i del candidat **7** per a la 9,9, hem aconseguit identificar l'origen de la hipertròfia de les seves trajectòries de cerca: les recursions de **RE-MEMBER** són amnèsiques i la seva miopia sols els permet veure l'element de **R-LLL** corresponent a la casella que està trepitjant; si a partir d'un moment donat queda clar que, 20 caselles lliures més endavant, l'element que representa la casella és una llista amb 9 elements **nil**, el node actual de l'arbre recorregut per **RE-MEMBE**R no rep cap

informació sobre aquest fet ni, per conseqüent, de que estem a punt d'entrar en un carrerò sense sortida, sinó que l'haurà de recórrer (amb l'agravant que no parlem d'un recorregut lineal sinó d'una branca amb els seus branquillons derivats) fins a trepitjar la casella-fulla (tantes vegades com branquillons condueixin allà), iniciant el retorn amb un fotimer de retrocessos fins a situar-se en la posició de partença i provar un altre candidat 1... 9. Sort que en aquest cas, sabent quin és el mal, no costa gens posar-hi un remei que, contra allò que pot semblar a primera vista, no comporta un alentiment perceptible: n'hi haurà prou a comprovar, després de l'actualització de **R-9*9** i **R-LLL**, si alguna llista (fila) d'aquest últim contè elements com el descrit i, si és així, obviar la crida autorecursiva. Com venim fent amb les versions adaptades de **RE-MEMBER** (per facilitar la continuïtat, seguim amb les variables no declarades que després d'interrompre l'execució ens permeten capturar la informació relativa als retrocessos), subratllarem l'afegitó:

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))
                    (if (not OK)
                        (progn
                          (setq *NIV* (cadr R-P))
                          (cond ((< *NIV* 3) (setq *REQ-1* (1+ *REQ-1*)))
                                ((< *NIV* 6) (setq *REQ-2* (1+ *REQ-2*)))
                                (    T       (setq *REQ-3* (1+ *REQ-3*))))))))))))
  OK)
```

Com veureu de seguida, algun fruit hem recollit d'aquesta versió **RE-MEMBER**, però la disfunció principal encara subsisteix, així que prescindirem del primer i segon requadre 9×3 i anirem de dret al tercer, que era on es produïa el conflicte:

Exploració i emplenament de 10 caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà **2.214** + **310** + **215** = **2.739** retrocessos
- Si també omplim 5,7 (**4**) hi haurà **1.740** + **217** + **39** = **1.996** retrocessos
- Si també omplim 9,7 (**8**) hi haurà **1.605** + **221** + **25** = **1.851** retrocessos
- Si també omplim 2,8 (**4**) hi haurà **1.968** + **276** + **43** = **2.287** retrocessos
- Si també omplim 4,8 (**9**) hi haurà **1.782** + **229** + **33** = **2.044** retrocessos
- Si també omplim 6,8 (**8**) hi haurà **1.346** + **171** + **52** = **1.569** retrocessos
- Si també omplim 8,8 (**3**) hi haurà **1.044** + **167** + **39** = **1.250** retrocessos
- Si també omplim 1,9 (**9**) hi haurà **1.113** + **3.348** + **1.570** = **6.031** retrocessos
- Si també omplim 5,9 (**3**) hi haurà **876** + **6.760** + **660** = **8.296** retrocessos
- Si també omplim 9,9 (**2**) hi haurà **?** + **?** + **?** = **?** retrocessos

---

Sumant retrocessos per requadres: **?** + **?** + **?** = **?** retrocessos

Comparant aquesta taula amb la precedent, veiem una lleugera reducció a la majoria de valors i, pel que fa als de 1,9 i 5,9, una bona davallada. A la casella 9,9 els retrocessos del candidat 7 també s'han abaixat, però els del candidat **5** segueixen sent inabastables:

```
1> 114 +     22 +  3 =    139
2> 114 +     24 +  0 =    138
4> 137 +     22 +  7 =    166
5>   ? +      ? +  ? =      ?
6> 138 +    211 + 14 =    363
7> 137 + 23.254 +  1 = 23.392
T>   ? +      ? +  ? =      ?
```

De segur que darrera de tanta contumàcia s'hi ha d'amagar alguna singularitat ben diferent de les tractades fins ara i que passa desapercebuda quan se situa en el primer requadre 9×3 però que resulta letal quan passa al tercer. Disposar de la solució virtual que correspongués al valor **5** en posició 9,9 ens ajudaria molt a trobar-la, però és precisament la cerca d'aquesta solució el que se'ns escapa de les mans, fins el punt d'haver-nos de preguntar si n'hi ha o no, de solució: n'hi hauria prou a imaginar una cerca ben accidentada i plagada de ziga-zagues, en què s'arribés a l'última casella lliure però no per assignar-li l'únic valor possible sinó per descobrir que ja no en queda cap. De tota manera, tres capítols endavant veurem que un SUDOKU-PROBLEMA ben plantejat (és a dir, amb una única solució) ha de tenir 17 caselles plenes, si més no. Així que un de 10 com aquest o té diverses solucions o no en té cap, i costa de creure que entre tan poques pugui haver-hi incompatibilitat (tant més quan 9 d'elles formen part de la solució NATURMÍN). Doncs bé, hi ha una manera senzilla d'obtenir la solució que se'ns escapa per via directa: com que una SUDOKU-SOLUCIÓ ho segueix sent permutant els requadres 9×3, si desplacem cap avall les 10 caselles de forma que quedin en el requadre inferior i n'introduïm els valors en l'ordre habitual (a l'esquerra els teniu en negreta i en posició 9,3 hi figura el candidat **5** en comptes del valor **2** previst), el temps de processat es normalitza en totes les 10 caselles,

Exploració i emplenament de 10 caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**5**) hi haurà **1.993** + **285** + **183** = **2.461** retrocessos
- Si també omplim 5,1 (**4**) hi haurà **1.338** + **168** + **193** = **1.699** retrocessos
- Si també omplim 9,1 (**8**) hi haurà **882** + **122** + **105** = **1.109** retrocessos
- Si també omplim 2,2 (**4**) hi haurà **1.338** + **357** + **146** = **1.841** retrocessos
- Si també omplim 4,2 (**9**) hi haurà **594** + **332** + **221** = **1.147** retrocessos
- Si també omplim 6,2 (**8**) hi haurà **135** + **180** + **72** = **387** retrocessos
- Si també omplim 8,2 (**3**) hi haurà **14** + **142** + **88** = **244** retrocessos
- Si també omplim 1,3 (**9**) hi haurà **76** + **222** + **79** = **377** retrocessos
- Si també omplim 5,3 (**3**) hi haurà **11** + **164** + **89** = **264** retrocessos
- Si també omplim 9,3 (**5**) hi haurà **79** + **147** + **128** = **354** retrocessos

---

Sumant retrocessos per requadres: **6.460** + **2.119** + **1.304** = **9.883** retrocessos

inclosa l'última on, desglossats per candidats, els retrocessos s'hauran reduït a

1> 13 + 36 + 5 = 54
2> 11 + 30 + 8 = 49
4> 11 + 20 + 8 = 39
5> 15 + 21 + 3 = 39
6> 11 + 20 + 52 = 83
7> 18 + 20 + 52 = 90
T> **79** + **147** + **128** = **354**

Fixeu-vos que, si en lloc d'emplenar la casella 9,3 amb un **5**, ho haguéssim fet amb un **2** (el valor que originalment se li assignava), haurien resultat una taula i un desglossament idèntics: els retrocessos es produeixen en activar la casella (i, ja ho hem dit abans, responen a tots els candidats compatibles en primera instància), de forma que el valor escollit només influirà en les caselles que activem després; així doncs, en l'última hi posem un valor (**5**) més que res per raons estètiques, per quadrar la taula, però igualment li escauria qualsevol altre valor compatible.

8 9 7	5 1 4	3 6 2	**9** 7 8	2 **3** 1	6 4 **5**
3 6 4	8 9 2	5 7 1	6 **4** 1	**9** 5 **8**	2 **3** 7
2 1 5	3 7 6	8 9 4	**5** 3 2	6 **4** 7	9 1 **8**
7 8 9	1 2 3	4 5 6	8 9 7	5 1 4	3 6 2
4 5 6	7 8 9	1 2 3	3 6 4	8 9 2	5 7 1
1 2 3	4 6 5	7 8 9	2 1 5	3 7 6	8 9 4
9 7 8	2 **3** 1	6 4 **5**	7 8 9	1 2 3	4 5 6
6 **4** 1	**9** **5** **8**	2 **3** 7	4 5 6	7 8 9	1 2 3
5 3 2	6 **4** 7	9 1 **8**	1 2 3	4 6 5	7 8 9

9 7 8	5 **3** 1	6 4 **2**	8 9 7	2 1 4	3 6 5
6 **4** 2	**9** 7 **8**	5 **3** 1	3 6 5	8 9 7	2 1 4
5 3 1	6 **4** 2	9 7 **8**	2 1 4	3 6 5	8 9 7
8 9 7	2 1 4	3 6 5	**9** 7 8	5 **3** 1	6 4 **2**
3 6 5	8 9 7	2 1 4	6 **4** 2	**9** 7 **8**	5 **3** 1
2 1 4	3 6 5	8 9 7	**5** 3 1	6 **4** 2	9 7 **8**
7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9

Destaquem el fet que, just després d'emplenar 9,3 (**5**), **ACTUALITZA-PLUS** emplenarà automàticament 5,2 (**5**), circumstància que ja ens anuncia de quin mal hem de morir: aquest valor és incompatible amb la casella 5,1 (**5**) de NATURMÍN, que després de la permutació circular de requadres 9×3 correspon a la posició 5,4. Veiem-ho si no: si ara passem a la casella 1,4 i seguim cap a la dreta i cap amunt, triant sempre

el valor més baix (patró d'emplenament NATURMÍN), i en acabada la 9ª fila tornem a la 1ª, 2ª i 3ª per emplenar les caselles buides amb el mateix criteri, resultarà la solució situada a l'esquerra. Aclarirem que, partint de les 11 caselles plenes (caràcters en negreta), hem anat emplenant la resta (caràcters normals) en l'ordre que tenien aquestes caselles abans de desplaçar el tercer requadre a baix de tot, el primer al mig i el segon a dalt. Al mig teniu la mateixa solució, restituint-li aquestes posicions primitives, i a la dreta hem repetit la solució NATURMÍN perquè destaquin bé (caselles subratllades en les dues) els canvis respecte a l'anterior com a conseqüència de la substitució de valors en la casella 9,9 (**2** per **5**).

El que ens interessa descobrir és per què, tenint els sudokus de l'esquerra i del centre el mateix nombre de solucions (que es corresponen biunívocament mitjançant la mateixa permutació de requadres 9×3, com les representades), a l'hora de provar **RE-MEMBER** el valor **5** en la 10ª casella activada, en el primer (9,3) n'hem trobat de seguida una i en l'altre (9,9) l'espera es prolongava tant que no hem tingut paciència per veure què passava amb aquest candidat. De fet el nostre era el **2**, que no ha trigat gens, i no podem passar per alt l'absurditat d'un contratemps que es pot produir altres vegades: a efectes pràctics, haver d'esperar a que desfilin tots els candidats per **RE-MEMBER** pot sortir molt car, i el cas present n'és un bon exemple. En algun moment haurem d'estudiar la possibilitat d'interrompre la cerca quan el candidat que l'usuari té *in mente* ja hagi estat validat, per tal que pugui confirmar-lo sense més dilació.

Quedi constància del compromís i recuperem el fil del discurs per desentranyar on rau la gran diferència de temps de procés entre els dos sudokus de què tractàvem. Abans que res cal aclarir que la representació del primer no es correspon amb la taula de retrocessos, que només enregistra l'emplenament virtual de la casella 9,3 amb el valor **5**, a càrrec de **RE-MEMBER** (que com sempre segueix el patró NATURMÍN), després d'haver emplenat de debò les altres 9 caselles en negreta. Tanmateix, la del segon sí que respon a les dues situacions (ho sabem per via deductiva, perquè fins que no superem l'escull de la casella 9,9 no ho podrem veure materialitzat): d'una banda la solució que obtindríem si, després d'emplenar les 10 caselles en negreta del requadre superior (a les quals **ACTUALITZA-PLUS** afegiria una 11ª, 5,8), anéssim a l'inici i emplenèssim les altres 70, seguint l'ordre i criteri NATURMÍN; de l'altra, l'emplenament virtual de la casella 9,9 amb el valor **5**, a càrrec de **RE-MEMBER**, després d'haver emplenat manualment les altres 9 caselles en negreta. En aquesta segona situació és on es produeix el conflicte, òbviament responsable de que tampoc pugui donar-se la primera, i la mare dels ous és que aquí encara no hi ha intervingut **ACTUALITZA-PLUS**: nosaltres coneixem la solució i la necessària presència d'un altre **5** a la casella 5,8 però, com que **RE-MEMBER** no sap res d'això, començarà l'emplenament creient que sobre la casella 5,1 no hi pesa més restricció que evitar els valors **3** i **4** presents a la mateixa columna; per acabar-ho d'adobar, la casella responsable de tot això es troba a l'altra banda de l'escaquer, 7 més amunt i, com que **RE-MEMBER** progressa pel procediment d'assaig i error (ras i curt, 10 passos endavant i 9 enrera), l'itinerari es farà més llarg que un dia sense pa. Circumstància, aquesta última, que no passa en el cas de l'esquerra, on la casella predeterminada és 5,2 i **RE-MEMBER** no trigarà a provar-hi el valor **5**. Així que, per confirmar aquest diagnòstic comparant les trajectòries impulsades per **RE-MEMBER** en un i altre cas, haurem de procurar-nos l'última dada que faltava: la representació de l'emplenament virtual de 9,3 amb un **5**, després d'haver emplenat manualment les altres 9 caselles en negreta, que correspon a l'última taula de retrocessos i és complementària de la que teníem a l'esquerra (a diferència de la del mig en què, repetim'ho un cop més, coincidien representacions d'emplenament real i virtual). La teniu en primer lloc, i és que, per tenir-les totes dues a mà, repetirem en el centre la de les 10 caselles ocupades en posició primitiva i, per torna, afegirem a la dreta una tercera configuració, fruit d'un plantejament erroni:

8 9 5	6 7 4	3 2 1	**9** 7 8	**2** **3** 1	6 4 **5**	9 4 1	6 7 8	2 3 **5**
6 7 1	8 2 3	9 5 4	6 **4** 1	**9** 5 **8**	2 **3** 7	8 7 5	9 3 2	6 1 4
4 3 2	5 9 1	6 8 7	**5** 3 2	6 **4** 7	9 1 **8**	6 3 2	5 1 4	9 7 8
7 8 9	4 1 2	5 6 3	8 9 7	5 1 4	3 6 2	5 9 8	2 4 7	3 6 1
3 5 6	7 8 9	4 1 2	3 6 4	8 9 2	5 7 1	3 6 7	8 9 1	5 4 2
1 2 4	3 6 5	8 7 9	2 1 5	3 7 6	8 9 4	2 1 4	3 6 5	8 9 7
9 6 8	1 **3** 7	2 4 **5**	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6
2 **4** 7	**9** 5 **8**	1 **3** 6	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3
5 1 3	2 **4** 6	7 9 **8**	1 2 3	4 6 5	7 8 9	1 2 3	4 5 6	7 8 9

A risc de semblar reiteratius, descriurem aquestes tres representacions i les tres precedents (les dues diferents, perquè la del mig està repetida) perquè és del tot imprescindible tenir les idees clares i hi ha matisos que, si no es perceben, ens poden dur a equívocs:

- Penúltim tercet, a la dreta (i antepenúltim, al mig): la solució NATURMÍN. Per definició, representa tant la solució que resulta d'emplenar totes les caselles en l'ordre i amb el criteri que hem acordat anomenar NATURMÍN, com l'emplenament virtual de qualsevol de les 81 caselles (estant les altres buides o plenes amb els valors anotats), a càrrec de **RE-MEMBER**, quan el valor candidat que es prova és el que finalment s'adopta (en particular, l'emplenament virtual de la casella 9,9 amb el valor **2**).
- Últim (i penúltim) tercet, al mig, torna a aparèixer més endavant: la solució obtenible introduint les 10 caselles en negreta del requadre 9×3 superior, en l'ordre 1,7 (**5**), 5,7 (**4**), 9,7 (**8**), 2,8 (**4**), 4,8 (**9**), 6,8 (**8**), 8,8 (**3**), 1,9 (**9**), 5,9 (**3**) i 9,9 (**5**), passant després a la casella 1,1 i, des d'aquí i en l'ordre i criteri NATURMÍN, anant emplenant la resta de caselles, tot i que igualment pot representar l'emplenament virtual de la casella 9,9 amb el valor **5**, a càrrec de **RE-MEMBER**, havent emplenat ja les 9 caselles en negreta precedents. Ni en una ni en l'altra accepció ha estat possible de conèixer-los directament (per la llarga espera que volem superar) i per això ha calgut recórrer a l'artifici d'obtenir la solució que ve a continuació i deduir-ne la present. El nombre de retrocessos en l'exploració **RE-MEMBER**, en activar la 10ª casella en negreta (9,9), no s'ha pogut determinar però de segur que s'eleva a molts milions, si tenim en compte que el temps d'espera del candidat **6** a la casella 1,9 abans de l'última millora no arribava a 11 minuts, representant 1.815.690 retrocessos, i que el candidat **5** a 9,9 ens ha mantingut més 6 hores a l'espera sense treure'n res.
- Penúltim tercet, a l'esquerra: la solució obtinguda introduint les 10 caselles en negreta del requadre 9×3 inferior (les de l'apartat precedent, corregudes 6 caselles cap avall), seguidament totes les dels altres dos requadres i finalment les que encara restaven lliures en el primer, sempre en l'ordre habitual (cap a la dreta i cap amunt (a destacar que els valors en caràcters normals són els més baixos que ens oferirà **RE-MEMBER** en cada cas, i que el valor **5** a 5,2 ja l'haurà posat **ACTUALITZA-PLUS** automàticament). El nombre de retrocessos en l'exploració **RE-MEMBER**, en activar la 10ª casella en negreta (9,3), s'ha detallat en l'última taula, amb un total de 9.883.
- Últim tercet, a l'esquerra, torna a aparèixer més endavant: la solució obtinguda introduint les 10 caselles en negreta del requadre 9×3 inferior (les mateixes de l'apartat precedent, inclòs el **5** que col·loca **ACTUALITZA-PLUS** automàticament a 5,2), passant després a 1,1 i, des d'aquí i en l'ordre i criteri NATURMÍN, anant emplenant la resta de caselles. Igualment pot representar l'emplenament virtual de la casella 9,3 amb el valor **5**, a càrrec de **RE-MEMBER**, havent emplenat ja les 9 caselles en negreta precedents, però en aquest cas no es produirà emplenament automàtic a 5,2 i si **RE-MEMBER** hi acaba assignant el valor **5** això serà després d'algunes giragonses que tindrem ocasió de seguir. El nombre de retrocessos en l'exploració **RE-MEMBER**, en activar la 10ª casella en negreta (9,3), és el mateix de l'apartat precedent, amb un total de 9.883.
- Últim tercet, a la dreta: la solució obtinguda assignant un **5** a la casella 9,9, passant després a la casella 1,1 i, des d'aquí i en l'ordre i criteri NATURMÍN, anant emplenant la resta de caselles. Hem portat aquí aquest sudoku en previsió que a algú se li acudís com alternativa avantatjosa a la via indirecta que hem seguit per obtenir la solució descrita en el segon apartat (treure-la de la del primer), tot pensant que ens havíem complicat la vida innecessàriament. És cert que aquest sudoku s'obté amb molt pocs retrocessos (3.692, com podeu veure a la taula *Exploració caselles 3r requadre amb l'escaquer buit*, emplenant només la casella 9,9, tant se val que ho fem amb **2**, **5** o qualsevol altre valor compatible) però no ho és que doni el mateix resultat, com podeu comprovar comparant els dos sudokus (centre i dreta, amb les diferències subratllades). L'equívoc pot venir de suposar que, com **RE-MEMBER** segueix una ruta exploratòria de patró NATURMÍN i els 9 primers valors emplenats en un (negreta) però no a l'altre no han de ser cap obstacle perquè tot es produeixi igual en ambdós casos, en ser al cap i a la fi caselles de la solució NATURMÍN, el resultat serà idèntic. Presumpció falsa (tot i que digna d'un sofista professional), perquè quan **RE-MEMBER** explori 9,9 amb un **2** sí que l'emplenament virtual es produirà com la solució NATURMÍN, però quan ho faci amb un **5** no: la primera diferència ja es produirà a la casella 3,5, on hi col·locarà un **7** en comptes d'un **5**, i quan arribi al requadre 9×3 superior només 9,7 (**8**), 4,8 (**9**) i 1,9 (**9**) coincidiran amb les caselles emplenades al mig.

Insistim en que, de les dues interpretacions de la quarta i segona representació (esquerra i centre), ens interessa la segona: considerar l'emplenament virtual amb què la **RE-MEMBER** estableix la viabilitat del candidat **5** en 9,3 (esquerra) i en 9,9 (centre), per detectar alguna diferència que justifiqui dues respostes tan dispars en sudokus que responen al mateix principi de generació NATURMÍN a partir de nou caselles prèviament emplenades, pel simple fet d'haver-se desplaçat verticalment 6 posicions. En el primer cas, l'emplenament de 5,2 amb un **5** es produeix tan aviat que es pot reproduir manualment la trajectòria d'exploració, i a partir d'aquí el de la resta de caselles és bufar i fer ampolles. Però obviament no passa el mateix amb el segon, raó per la qual haurem de readaptar els dispositius que treien per pantalla de text el nombre de retrocessos i permetien conèixer l'ocupació virtual (**R-9*9** i **R-LLL**) en un moment determinat (així havíem descobert que l'existència de caselles que ja no admetien cap valor, molt per davant de l'actual, era una de les causes del retard), per realitzar sondejos en situacions que jutgem especialment significatives i poder apreciar si hi ha alguna dinàmica progressiva subjacent a la repetició cíclica de seqüències en el procés d'emplenament virtual. Però fins i tot amb aquestes eines ho tindríem més difícil si no coneguéssim les solucions per endavant: ja se sap que, per estar puntualment informats de la situació de tots els participants d'una gimcana, cal conèixer el recorregut i els punts de control.

Començarem per l'emplenament virtual representat a l'esquerra de l'últim tercet, amb **RE-MEMBER** verificant a 9,3 la idoneïtat del candidat **5**. Ja hem comentat prou que allò que diferencia aquest procés del que s'esdevé quan, poc després, havent desfilat tots els candidats aptes i havent seleccionat el valor **5**, **ACTUALITZA-PLUS** n'assigna un altre a la casella 5,2, és que aquesta funció deixa d'intervenir quan és reclamada des de **RE-MEMBER**, perque l'argument **REAL** és nul. Però en aquest cas les conseqüències de la no intervenció d'**ACTUALITZA-PLUS** són fàcilment superables, perquè com veurem el conflicte es concentra en les 4 primeres files (2ª, 3ª i 4ª) i, malgrat que **RE-MEMBER** actua com una força cega que ignora la predeterminació de la casella 5,2, en ser només la 9ª lliure en la ruta de cerca, no trigarà massa a assolir el seu destí. La seqüència representada a continuació mostra tres estats successius del primer requadre (per no fer-ho massa extens, hem passat per alt els retrocessos encara més locals), en l'últim dels quals ja haurà adoptat el valor **5**:

9 0 0	0 **3** 0	0 0 **5**
0 **4** 0	**9** 0 **8**	0 **3** 0
5 1 2	6 **4** 7	9 X **8**

9 0 0	0 **3** 0	0 0 **5**
2 **4** 6	**9** 1 **8**	X **3** 0
5 1 3	2 **4** 6	7 9 **8**

9 6 8	1 **3** 7	2 4 **5**
2 **4** 7	**9** 5 **8**	1 **3** 6
5 1 3	2 **4** 6	7 9 **8**

A partir d'aquí, ampliem les representacions a l'escaquer 9×9 complet: els quatre primers fotogrames descriuen (saltant-nos de nou retrocessos més locals) els pocs entrebancs que, seguint el criteri NATURMÍN, ens el deixen completament emplenat.

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
4 3 2	5 7 1	6 8 X

0 0 0	0 0 0	0 0 0
3 5 6	4 1 2	X 0 0
1 2 4	3 6 5	8 7 9

7 8 9	4 1 2	3 5 X
3 5 6	7 8 9	4 1 2
1 2 4	3 6 5	8 7 9

7 8 9	4 1 2	5 6 3
3 5 6	7 8 9	4 1 2
1 2 4	3 6 5	8 7 9

9 6 8	1 **3** 7	2 4 **5**
2 **4** 7	**9** 5 **8**	1 **3** 6
5 1 3	2 **4** 6	7 9 **8**

9 6 8	1 **3** 7	2 4 **5**
2 **4** 7	**9** 5 **8**	1 **3** 6
5 1 3	2 **4** 6	7 9 **8**

9 6 8	1 **3** 7	2 4 **5**
2 **4** 7	**9** 5 **8**	1 **3** 6
5 1 3	2 **4** 6	7 9 **8**

8 9 5	6 7 4	3 2 1
6 7 1	8 2 3	9 5 4
4 3 2	5 9 1	6 8 7

9 0 0	0 **3** 0	0 0 **5**
0 **4** 0	**9** 0 **8**	0 **3** 0
5 0 0	0 **4** 0	0 0 **8**

9 7 8	2 **3** 1	6 4 **5**
6 **4** 1	**9** 5 **8**	2 **3** 7
5 3 2	6 **4** 7	9 1 **8**

7 8 9	4 1 2	5 6 3
3 5 6	7 8 9	4 1 2
1 2 4	3 6 5	8 7 9

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
2 0 0	0 0 0	0 0 0

8 9 7	5 1 4	3 6 2
3 6 4	8 9 2	5 7 1
2 1 5	3 7 6	8 9 4

9 6 8	1 **3** 7	2 4 **5**
2 **4** 7	**9** 5 **8**	1 **3** 6
5 1 3	2 **4** 6	7 9 **8**

7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9

7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	4 6 5	7 8 9

Aquesta trajectòria és una autèntica marxa triomfal, comparant-la amb el viacrucis que segons tots els indicis (incomplets, però tot i així aclaparadors) devia tenir lloc amb les ocupacions desplaçades 6 caselles amunt. Val la pena fer l'esforç de reconstruir-lo, ni que sigui a grans trets, per esbrinar què provoca que la durada del procés es dispari de manera inesperada. Per no obligar al lector a retrocedir massa fulls per localitzar l'emplenament virtual que abans hem determinat per via indirecta (últim tercet de solucions, al mig), el reproduirem aquí, aprofitant el buit deixat per la sèrie precedent (a baix, a la dreta). Entre els dos (a baix, al mig), teniu l'escaquer 9×9 tal i com el tindríem virtualment després d'emplenar les 9 primeres caselles i activar la 10ª, mentre **RE-MEMBER** hi comprova el valor **5**, just quan l'embranzida inicial ha completat el requadre 9×3 inferior i ocupat la primera casella del central. Igual que abans a la casella 5,2, la funció esmentada ignora que a la 5,8 li està predestinat un altre **5**, però així com en aquell cas l'emplenament virtual començava pel requadre on s'amagava aquest petit secret i la seva pròpia dinàmica resolia el problema en un tres i no res, ara res no s'oposarà en aquesta embranzida inicial a que la casella 5,1 adopti el valor **5**, de manera que, quan l'exploració **RE-MEMBER** arribi al requadre 9×3 superior no podrà situar cap **5** a la casella 5,8 (feina que **ACTUALITZA-PLUS** no ha pogut assumir) i quedarà bloquejat després d'emplenar 7,8 (on tampoc hi té cabuda el **5**), perquè la casella 9,8 ja no admetrà cap valor. L'única sortida serà retornar sobre el seus passos, però no fent marxa enrera de manera ininterrompuda, sinó en una penosa retirada zigzaguejant (ara, 9 passos endavant i 10 enrera), reculant fins la casella 5,1 i recomençant amb **6**. Però per quantificar en retrocessos aquest periple intrincat, ni que sigui aproximadament, millor que comencem per un cicle més curt.

La seqüència de 3×4 representacions pretén descriure el primer dels cicles que es van succeint, cadascun d'ells mantenint constant el primer requadre 9×3, i que es subdivideixen en quatre cicles corresponents als quatre valors que pot acceptar la casella 1,4: 2, 3, 6 i 8. Cadascun dels 4 cicles interns està representat per una fila de 3 representacions: la situació inicial (esquerra), la d'emplenament més avançat amb l'última versió de **RE-MEMBER** (centre) i la d'emplenament més avançat amb la penúltima (dreta); sota les dues primeres teniu els retrocessos deguts al conjunt de candidats, desglossats, com sempre, segons afectin al 1r, al 2n o al 3r requadre 9×3. Com que d'entrada pot estranyar que es torni a utilitzar una versió més lenta, ja superada, però sobretot que les representacions del mig no mostrin un emplenament que ja ha arribat a la 8ª fila, farem un aclariment que valdrà per ambdues qüestions: la penúltima versió de **RE-MEMBER**, obsoleta per ser més lenta, té tanmateix la particularitat de no aturar-se fins que no es troba que la següent casella lliure ja no admet assignació de valor, i funcionant d'aquesta manera sí que arriba a ocupar virtualment la casella 7,8, després d'haver emplenat la 5,8 amb un **7** (1ª i 4ª fila de representacions) o amb un **1** (2ª i 3ª fila), sense poder fer res amb la casella 9,8. De fet, la casella 5,8 també admetrà el valor **6** quan a 4,5, 5,5 i 6,5 no n'hi hagi cap, i ben segur que després de les que hem presentat deu haver-hi alguna remuntada amb aquest valor, però com que l'únic que es volia mostrar és que amb un primer requadre 9×3 com el que tenim s'arriba a emplenar 8,8 però d'aquí no es passa, ens hem limitat a les primeres incursions fins la posició esmentada per a cada valor de 1,4: teniu en compte que els tres primers casos (ens referim a les representacions de la dreta) s'han localitzat de forma immediata a partir dels precedents (les representacions del mig), ja que entre les situacions del mig i de la dreta de la 1ª fila no hi ha retrocessos, i entre les de la 2ª i 3ª fila gairebé tampoc, però el quart (4ª fila a la dreta) només ha estat possible de trobar-lo implementant en la penúltima versió de **RE-MEMBER** un dispositiu *ad hoc* i armant-se de paciència per esperar el resultat. El que no queda clar és el paper que hi juguen les representacions del mig: si acabem de confessar que precedeixen poc o molt les de la dreta, que a sobre tenen emplenada una fila més, ¿com podem afirmar que representen un màxim d'ocupació dintre del cicle intern? Doncs sí que el representen, però només pel que fa a les exploracions fetes amb l'última versió de **RE-MEMBER**, que per guanyar temps pren dreceres: recordeu que, per aturar-se no espera a adonar-se que la següent casella lliure ja no admet assignació de valor, sinó que detecta a distància qualsevol altra casella condemnada, s'atura i emprèn la retirada fins a trobar una ocasió més propícia, i funcionant d'aquesta manera aconsegueix reduir els retrocessos a les quantitats que figuren en negreta; en la versió anterior, en general les xifres serien més elevades, i és per aquesta raó (no barrejar recomptes que es basen en criteris diferents) que les representacions de la dreta no inclouen cap referència als retrocessos. Observeu que la punta de llança (l'últim emplenament en les representacions del mig, que en tots 4 cicles se situa en la casella 6,7) ja no penetra més perquè dues posicions més amunt la casella 8,7 quedarà sense assignació possible, fem el que fem.

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 0 0 0 **4** 0 0 0 **8**

0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **47** + **0**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 0 0 0 **4** 0 0 0 **8**

0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **105.971** + **10.624**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 0 0 0 **4** 0 0 0 **8**

0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **324.672** + **24.704**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 0 0 0 **4** 0 0 0 **8**

0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **393.934** + **28.160**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 0 0 0 **4** 0 0 0 **8**

0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0

7 8 9 1 2 3 5 4 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

141 + **640.222** + **42.240**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 0 0 0 **4** 0 X 0 **8**

8 9 7 5 1 2 3 6 4
3 6 4 8 9 7 5 1 2
2 1 5 3 6 4 8 9 7

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **2.677** + **191**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 3 1 2 **4** 7 X 0 **8**

8 9 4 3 7 1 5 6 2
6 7 5 8 9 2 3 1 4
3 1 2 5 6 4 8 9 7

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **106.039** + **10.628**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 3 1 2 **4** 7 X 0 **8**

8 9 2 5 6 4 3 1 7
3 7 4 8 9 1 5 6 2
6 1 5 3 7 2 8 9 4

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **330.026** + **25.193**

9 0 0 0 **3** 0 0 0 **5**
0 **4** 0 **9** 0 **8** 0 **3** 0
5 3 1 2 **4** 7 X 0 **8**

6 9 5 8 7 2 3 1 4
3 7 4 5 9 1 8 6 2
8 1 2 3 6 4 5 9 7

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **393.971** + **28.160**

9 0 0 0 **3** 0 0 0 **5**
2 **4** 7 **9** 1 **8** 6 **3** X
5 3 1 2 **4** 7 9 7 **8**

8 9 4 3 7 1 5 6 2
6 7 5 8 9 2 3 1 4
3 1 2 5 6 4 8 9 7

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **106.039** + **10.628**

9 0 0 0 **3** 0 0 0 **5**
2 **4** 7 **9** 1 **8** 6 **3** X
5 3 1 2 **4** 7 9 7 **8**

8 9 2 5 6 4 3 1 7
3 7 4 8 9 1 5 6 2
6 1 5 3 7 2 8 9 4

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **330.026** + **25.193**

9 0 0 0 **3** 0 0 0 **5**
2 **4** 1 **9** 7 **8** 6 **3** X
5 3 7 6 **4** 2 9 1 **8**

6 9 5 8 1 7 3 4 2
3 7 4 2 9 5 8 6 1
8 1 2 3 6 4 5 9 7

7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
1 2 3 4 5 6 7 8 9

137 + **393.971** + **28.160**

Si a la pàgina precedent parlàvem d'una seqüència de 3×4=12 representacions per descriure el primer cicle, ¿què representa aquesta 13ª a l'esquerra?: doncs un nou cicle, on tornaran a succeir-se a la casella 1,4 el **2** (és la situació de l'esquerra), el **3,** el **6** i el **8.** Això vol dir que hem entrat en un cercle viciós? Ni de bon tros: en la dinàmica autorecursiva de **RE-MEMBER,** si pel que fa a les files 4ª a 9ª de l'escaquer 9×9 hem retornat al punt de partença és perquè els retrocessos ja han sobrepassat aquest llindar i de nou trepitgem el primer requadre 9×3, que ara acaba amb 5-4-6 en

lloc de 4-5-6, últim tercet de la 3ª fila que, al llarg del primer cicle, s'havia mantingut constant com la resta del requadre. El muntatge instal·lat a **RE-MEMBER** per seguir-li el rastre a l'exploració revela que, cada vegada que es repeteix el cicle referit a la successió dels valors **2**, **3**, **6** i **8** a la casella 1,4, els valors d'aquest tercet varien: després de 4-5-6 i 5-4-6, vénen 5-6-4 i 6-5-4. I, de nou, 4-5-6, 5-4-6, 5-6-4 i 6-5-4, definint un cicle de període més ampli, mentre el testimoni d'aquesta singular cursa de relleus passa al tercet precedent, on també comencen a succeir-se cíclicament els valors 1-2-3, 2-1-3, 3-1-2 i 3-2-1 ... ¿Cal esperar més (per ara cada cicle en conté 4 i, per tant, cada període és 4 cops més llarg) per adonar-se de per on van els trets? Ja és hora de començar a fer números i, abans de posar-nos-hi, volem deixar clar que l'esperit que ens ha de guiar no pot ser el de l'exactitud en el càlcul sinó el d'adquirir una noció aproximada de l'ordre de magnitud del nombre de retrocessos necessari per arribar a situar un **6** a la casella 5,1 i desbloquejar el camí cap a una progressió ja imparable (si no ininterrompuda, sí al menys no tan accidentada com la de l'etapa anterior) fins la posició 5,8 i d'aquí a l'emplenament virtual complet del sudoku. En aquest sentit, considerarem els canvis dintre del primer requadre a nivell de tercet, despreciant tant els **184** retrocessos inicials com els que hi hagi a cavall de l'esprint final, i suposarem que en tots els cicles externs el nombre de retrocessos associat a un cicle intern (els circumscrits al segon i tercer requadre 9×3) és, de mitjana, com el descrit i contabilitzat a la pàgina precedent.

Si el primer cicle que abasta els requadres 2 i 3 ha començat amb **184** retrocessos acumulats (**137** + **47** + **0**) i el segon amb **682.603** (**141** + **640.222** + **42.240**), dintre d'aquest cicle s'hauran produït **682.603** - **184** = **682.419** retrocessos, dels quals **4** s'han produït el el requadre 1 (**141** - **137**) i corresponen a la permutació de 4-5-6 a 5-4-6 i la resta en els requadres 2 i 3. Tot i que l'intercanvi de valors entre les caselles 7,3 i 8,3 farà que aquest segon cicle sigui diferent del primer, com hem anunciat suposarem que el nombre de retrocessos no serà molt diferent dels **682.419** del primer, ni els del tercer ni els del quart... Sempre considerarem el mateix valor, així que, acceptant la simplificació i sistematitzant l'exposició:

- **RE-MEMBER** retrocedirà fins el tercer tercet de la 3ª fila que, de les 6 permutacions 4-5-6, 4-6-5, 5-4-6, 5-6-4, 6-4-5 i 6-5-4, només pot formar les 4 subratllades (el **5** no és admissible a la 9ª columna). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{4 \times 682.419}$ retrocessos.
- **RE-MEMBER** retrocedirà fins el segon tercet de la 3ª fila que, de les 6 permutacions 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2 i 3-2-1, només pot formar les 4 subratllades (el **3** no és admissible a la 5ª columna). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{4^2 \times 682.419}$ retrocessos.
- **RE-MEMBER** retrocedirà fins el primer tercet de la 3ª fila que, de les 6 permutacions 7-8-9, 7-9-8, 8-7-9, 8-9-7, 9-7-8 i 9-8-7, només pot formar les 4 subratllades (el **9** no és admissible a la 1ª columna). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{4^3 \times 682.419}$ retrocessos.
- **RE-MEMBER** retrocedirà fins el tercer tercet de la 2ª fila que, de les 6 permutacions 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2 i 3-2-1, només pot formar les 4 subratllades (el **3** no és admissible a la 8ª columna). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{4^4 \times 682.419}$ retrocessos.
- **RE-MEMBER** retrocedirà fins el segon tercet de la 2ª fila que, de les 6 permutacions 7-8-9, 7-9-8, 8-7-9, 8-9-7, 9-7-8 i 9-8-7, només pot formar les 3 subratllades (el **9** no és admissible a la 4ª columna ni el **8** a la 6ª). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{3 \times 4^4 \times 682.419}$ retrocessos.
- **RE-MEMBER** retrocedirà fins el primer tercet de la 2ª fila que, de les 6 permutacions 4-5-6, 4-6-5, 5-4-6, 5-6-4, 6-4-5 i 6-5-4, només pot formar les 3 subratllades (el **5** no és admissible a la 1ª columna ni el 4 a la 2ª). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i

les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{3^2 \times 4^4 \times 682.419}$ retrocessos. Però encara hi ha més possibilitats: a diferència dels valors dels dos tercets precedents, determinats pel de l'esquerra i el de baix, aquest també admet els valors 7, 8 i 9, a condició que els tercets següents estiguin formats per 1, 2 i 3, i per 4, 5 i 6, respectivament; de fet, aquesta opció addicional equival a permutar senceres les files 2ª i 3ª, així que, en haver-les utilitzat també sense èxit, haurà acumulat $\mathbf{2 \times 3^2 \times 4^4 \times 682.419}$ retrocessos.

- **RE-MEMBER** retrocedirà fins el tercer tercet de la 1ª fila que, de les 6 permutacions 7-8-9, 7-9-8, 8-7-9, 8-9-7, 9-7-8 i 9-8-7, només pot formar les 4 subratllades (el **8** no és admissible a la 9ª columna). Com que amb cap d'aquestes tríades de valors no es podrà completar la 8ª fila i les exploracions sempre retrocediran fins el tercet, en acabar-les haurà acumulat $\mathbf{2 \times 3^2 \times 4^5 \times 682.419}$ retrocessos.
- **RE-MEMBER** retrocedirà fins el segon tercet de la 1ª fila, on amb la tríada original 4-5-6 es produirà l'únic cicle d'aquest nivell, perquè la següent a ser provada, 4-6-5, ja iniciarà la progressió (amb pocs retrocessos) que aconseguirà emplenar virtualment totes les caselles buides.

Així doncs, el nombre total de retrocessos (amb les reserves expressades abans) és $\mathbf{2 \times 3^2 \times 4^5 \times 682.419 = 12.578.347.008}$, xifra que fa pensar en dies d'espera, abans que en hores o minuts (recordeu que 1.815.690 retrocessos, amb l'antiquat equip de l'autor, gairebé representaven 11 minuts) i que justifica de sobres que ens prenem la molèstia d'evitar aquestes situacions traslladant a zona inferior de l'escaquer el contingut del requadre 9×3 més poblat.

Bé que, si el problema ve del **5** en 9,9, ¿per què hem dit abans que la posició de la casella actual era pràcticament irrellevant? Perquè no importa tant aquesta posició com la del forat que **ACTUALITZA-PLUS** hauria tapat si l'emplenament hagués estat real i no virtual, forat (5,8) que té més probabilitat de produir-se en el requadre 9×3 més poblat: en el cas present aquest requadre conté tant el forat com la casella actual, però no ha de ser així forçosament.

De tota manera, aniríem més segurs si trobéssim la manera de neutralitzar aquestes situacions específicament, sense confiar-ho tot en el dispositiu que ens haurà de deixar ordenats els requadres 9×3, a partir del que tingui més caselles plenes, de de més a menys en sentit ascendent. Perquè de vegades el forat pot situar-se en el requadre superior sense que aquest sigui el més ple, i no cal apartar-nos massa de l'últim exemple. De les 30 caselles aïllades que havíem destacat en negreta quan pàgines enrera presentàvem la solució NATURMÍN, 10 en cada requadre, fins ara ens havíem dedicat a omplir-ne només una o les 10 d'un mateix requadre, com en aquest exemple (l'analisi se situava en l'exploració prèvia a l'emplenament de l'última). El que farem ara és emplenar 18 d'aquestes 30 caselles, 6 en cada requadre 9×3, partint de 1,1, seguint l'alineació sinuosa que (tres pàgines endavant) mostra en negreta la representació de l'esquerra i acabant a 9,9. En activar aquesta última casella, igual que abans, apareixeran en el caseller 3×3 els candidats **1**, **2** i **4**, després hi haurà una llarga pausa (bé, no tan llarga com abans, perquè ara "només" serà d'uns 50 minuts) passada la qual acabaran de sortir els candidats **5**, **8** i **9**, i commutant a pantalla de text veurem quants retrocessos s'han produït i comprovarem que el conflicte desfermat pel **5** ha esclatat de nou, atenuat en aquest cas per la menor quantitat de caselles desocupades:

```
     1>     3 +         3 +         5 =        11
     2>     4 +         2 +         0 =         6
     4>     4 +         9 +         9 =        22
     5> 3.899 + 3.111.621 + 5.187.614 = 8.303.134
     8>     4 +         3 +        11 =        18
     9>     5 +        29 +        55 =        89
     T> 3.919 + 3.111.667 + 5.187.694 = 8.303.280
```

En aquest cas el dispositiu d'ordenació de requadres 9×3 no hauria pogut salvar la situació perquè, com que tots tres tenen 6 caselles emplenades (això comptant com a ocupada la casella 9,9, que en realitat només ho està virtualment), no jutjaria necessari d'intervenir.

Així doncs, no hi haurà més remei que introduir en **RE-MEMBER** una modificació que no només estarà en el cap de tothom a hores d'ara (més d'un cop hem comentat per què **ACTUALITZA-PLUS** no actua quan **ACTUALITZA** és reclamat des de **RE-MEMBER**) sinó

que serà fàcilíssim d'implementar, atès que la primera crida a aquesta funció la tenim diferenciada mitjançant l'expressió **(if INI (setq ... INI ()))**, que passarem a **(if INI (setq ... INI T))**, i llestos. Fora d'això, únicament haurem de tocar la funció **PLUS-N->P**, afegint-hi els subratllats:

```
(defun PLUS-N->P ()
  (setq A-N N A-P K PLUS T)
  (if (and REAL (not INI)) (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))))
```

Si després aquests minsos canvis tornem a emplenar les 18 caselles, el resultat a l'hora d'activar 9,9 serà molt més satisfactori:

```
1>  3 +   3 +   5 =  11
2>  4 +   2 +   0 =   6
4>  4 +   9 +   9 =  22
5>  4 + 110 + 165 = 279
8>  4 +   3 +  11 =  18
9>  5 +  29 +  55 =  89
T> 24 + 156 + 245 = 425
```

En vista de l'èxit, repetirem les exploracions amb 10 emplenaments per requadre, però no les limitarem a l'últim sinó a tots tres, tant per apreciar-ne els guanys secundaris com per tornar a la qüestió inicial, ara ja sense distorsions afegides: els retrocessos originats per una ocupació similar en cadascun del tres requadres.

Exploració i emplenament de 10 caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà	**1.993**	+	**285**	+ **183**	= **2.461**	retrocessos
- Si també omplim 5,1 (**5**) hi haurà	**1.338**	+	**147**	+ **118**	= **1.603**	retrocessos
- Si també omplim 9,1 (**9**) hi haurà	**882**	+	**252**	+ **48**	= **1.182**	retrocessos
- Si també omplim 2,2 (**5**) hi haurà	**1.338**	+	**452**	+ **147**	= **1.937**	retrocessos
- Si també omplim 4,2 (**7**) hi haurà	**594**	+	**337**	+ **68**	= **999**	retrocessos
- Si també omplim 6,2 (**9**) hi haurà	**135**	+	**242**	+ **40**	= **417**	retrocessos
- Si també omplim 8,2 (**2**) hi haurà	**14**	+	**217**	+ **48**	= **279**	retrocessos
- Si també omplim 1,3 (**7**) hi haurà	**12**	+	**251**	+ **28**	= **291**	retrocessos
- Si també omplim 5,3 (**2**) hi haurà	**13**	+	**201**	+ **37**	= **251**	retrocessos
- Si també omplim 9,3 (**6**) hi haurà	**33**	+	**321**	+ **21**	= **375**	retrocessos
Sumant retrocessos per requadre:	**6.352**	+	**2.705**	+ **738**	= **9.795**	retrocessos

Exploració i emplenament de 10 caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà	**2.214**	+	**323**	+ **351**	= **2.888**	retrocessos
- Si també omplim 5,4 (**6**) hi haurà	**1.771**	+	**213**	+ **39**	= **2.023**	retrocessos
- Si també omplim 9,4 (**7**) hi haurà	**1.644**	+	**133**	+ **23**	= **1.800**	retrocessos
- Si també omplim 2,5 (**6**) hi haurà	**1.968**	+	**191**	+ **58**	= **2.217**	retrocessos
- Si també omplim 4,5 (**8**) hi haurà	**1.782**	+	**218**	+ **44**	= **2.044**	retrocessos
- Si també omplim 6,5 (**7**) hi haurà	**1.571**	+	**76**	+ **75**	= **1.722**	retrocessos
- Si també omplim 8,5 (**1**) hi haurà	**1.128**	+	**23**	+ **33**	= **1.184**	retrocessos
- Si també omplim 1,6 (**8**) hi haurà	**1.204**	+	**27**	+ **54**	= **1.285**	retrocessos
- Si també omplim 5,6 (**1**) hi haurà	**798**	+	**60**	+ **39**	= **897**	retrocessos
- Si també omplim 9,6 (**5**) hi haurà	**806**	+	**29**	+ **25**	= **860**	retrocessos
Sumant retrocessos per requadr:	**14.286**	+	**1.293**	+ **741**	= **16.920**	retrocessos

Exploració i emplenament de 10 caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**2.214**	+	**310**	+ **215**	= **2.739**	retrocessos
- Si també omplim 5,7 (**4**) hi haurà	**1.740**	+	**217**	+ **39**	= **1.996**	retrocessos
- Si també omplim 9,7 (**8**) hi haurà	**1.605**	+	**221**	+ **25**	= **1.851**	retrocessos
- Si també omplim 2,8 (**4**) hi haurà	**1.968**	+	**276**	+ **43**	= **2.287**	retrocessos
- Si també omplim 4,8 (**9**) hi haurà	**1.782**	+	**229**	+ **33**	= **2.044**	retrocessos
- Si també omplim 6,8 (**8**) hi haurà	**1.346**	+	**171**	+ **52**	= **1.569**	retrocessos
- Si també omplim 8,8 (**3**) hi haurà	**1.044**	+	**167**	+ **39**	= **1.250**	retrocessos
- Si també omplim 1,9 (**9**) hi haurà	**1.113**	+	**3.348**	+ **1570**	= **6.031**	retrocessos
- Si també omplim 5,9 (**3**) hi haurà	**876**	+	**6.760**	+ **660**	= **8.296**	retrocessos
- Si també omplim 9,9 (**2**) hi haurà	**777**	+	**23.598**	+ **42**	= **24.417**	retrocessos
Sumant retrocessos per requadr:	**14.465**	+	**35.297**	+ **2718**	= **52.480**	retrocessos

El detall de la casella 9,9 ara és:

```
     1> 114 +     22 +   3 =    139
     2> 114 +     24 +   0 =    138
     4> 137 +     22 +   7 =    166
     5> 137 +     65 +  17 =    219
     6> 138 +    211 +  14 =    363
     7> 137 + 23.254 +   1 = 23.392
     T> 777 + 23.598 +  42 = 24.417
```

Havent arribat a aquest punt, ens preguntem si encara val la pena que permutem els requadres 9×3 per deixar-los ordenats de més a menys població en sentit ascendent. D'una banda sembla que sí perquè, malgrat els mecanismes correctors incorporats, se segueixen complint els principis generals que justifiquen aquesta ordenació:

- Una mateixa quantitat de caselles plenes (10) dóna lloc a més retrocessos si es concentra en el tercer requadre (52.480) que si ho fa en el segon (16.920), i en aquest últim cas més que si es concentra en primer requadre (9.795).
- Sigui quin sigui el requadre on es concentren els emplenaments, els retrocessos a què donen lloc no es reparteixen uniformement sinó que se'n produeixen més en el primer requadre que en el segon, i en aquest més que en el tercer.
- En tots els requadres s'esdevé que, quantes més caselles haguem emplenat, menys retrocessos comportaran els nous l'emplenaments.

(En el cas que ens ocupa, veureu que l'últim tercet del requadre superior és una excepció pel que fa als dos últimes principis.)
Però de l'altra, l'existència d'excepcions i el fet que les tendències no siguin molt marcades (amb això volem dir que la constatació A>B no es manifesti en forma de valors molt elevats de A/B) ens en podria fer desistir, en la sospita que la durada addicional que representés la permutació de requadres podria no compensar el temps guanyat en les exploracions **RE-MEMBER**. Però abans de prendre una decisió precipitada serà bo completar els assajos per requadres realitzats fins el moment (1 casella primer i 10 després) amb el seu emplenament total (ara les 27 caselles) i amb la intervenció sobre més d'un requadre (ni que sigui per veure, com abans en l'alineació sinuosa des de 1,1 fins a 9,9, si la reordenació de requadres pot o no salvar sempre la situació). Veiem-ho.

En el requadre inferior no hi haurà cap problema, amb uns resultats previsibles i que confirmen les tendències que fins ara apuntàvem:

Exploració i emplenament de les caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà	**1.993**	+	**285**	+ **183**	= **2.461**	retrocessos
- Si també omplim 2,1 (**2**) hi haurà	**1.747**	+	**242**	+ **166**	= **2.155**	retrocessos
- Si també omplim 3,1 (**3**) hi haurà	**1.501**	+	**210**	+ **21**	= **1.732**	retrocessos
- Si també omplim 4,1 (**4**) hi haurà	**846**	+	**144**	+ **35**	= **1.025**	retrocessos
- Si també omplim 5,1 (**5**) hi haurà	**600**	+	**112**	+ **63**	= **775**	retrocessos
- Si també omplim 6,1 (**6**) hi haurà	**354**	+	**311**	+ **33**	= **698**	retrocessos
- Si també omplim 7,1 (**7**) hi haurà	**738**	+	**89**	+ **9**	= **836**	retrocessos
- Si també omplim 8,1 (**8**) hi haurà i també s'omple 9,1 (**9**)	**492**	+	**59**	+ **6**	= **557**	retrocessos
- Si també omplim 1,2 (**4**) hi haurà	**846**	+	**381**	+ **128**	= **1.355**	retrocessos
- Si també omplim 2,2 (**5**) hi haurà	**600**	+	**347**	+ **66**	= **1.013**	retrocessos
- Si també omplim 3,2 (**6**) hi haurà	**354**	+	**318**	+ **72**	= **744**	retrocessos
- Si també omplim 4,2 (**7**) hi haurà	**363**	+	**82**	+ **9**	= **454**	retrocessos
- Si també omplim 5,2 (**8**) hi haurà	**72**	+	**58**	+ **6**	= **136**	retrocessos
- Si també omplim 6,2 (**9**) hi haurà	**12**	+	**30**	+ **3**	= **45**	retrocessos
- Si també omplim 7,2 (**1**) hi haurà	**0**	+	**116**	+ **12**	= **128**	retrocessos
- Si també omplim 8,2 (**2**) hi haurà i també s'omple 9,2 (**3**)	**0**	+	**60**	+ **7**	= **67**	retrocessos
- Si també omplim 1,3 (**7**) hi haurà	**0**	+	**89**	+ **9**	= **98**	retrocessos
- Si també omplim 2,3 (**8**) hi haurà i també s'omple 3,3 (**9**)	**0**	+	**59**	+ **6**	= **65**	retrocessos
- Si també omplim 4,3 (**1**) hi haurà	**0**	+	**76**	+ **16**	= **92**	retrocessos
- Si també omplim 5,3 (**2**) hi haurà i també s'omple 6,3 (**3**)	**0**	+	**57**	+ **6**	= **63**	retrocessos
- Si també omplim 7,3 (**4**) hi haurà	**0**	+	**107**	+ **10**	= **117**	retrocessos
- Si també omplim 8,3 (**5**) hi haurà i també s'omple 9,3 (**6**)	**0**	+	**60**	+ **6**	= **66**	retrocessos
Sumant retrocessos per requadres:	**10.518**	+	**3.292**	+ **872**	= **14.682**	retrocessos

```
0 0 0   0 0 0   0 0 5     0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0
0 4 0   9 0 8   0 3 0     0 0 0   0 0 0   0 0 0     6 4 2  (1)7 8   X 0 0
5 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     5 3 1   6 4 2   9 7 8

8 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     8 9 7   X 0 0   0 0 0
0 6 0   8 0 7   0 1 0     3 6 5  (1)9 8   X 0 0     3 6 5   8 9 7   2 1 4
0 0 0   0 0 0   0 0 7     2 1 4   3 6 5   8 9 7     2 1 4   3 6 5   8 9 7

0 0 0   0 0 0   0 0 6     7 8 9   2 1 3   4 5 6     7 8 9   2 1 3   4 5 6
0 5 0   7 0 9   0 2 0     4 5 6   7 8 9   1 2 3     4 5 6   7 8 9   1 2 3
1 0 0   0 0 0   0 0 0     1 2 3   4 5 6   7 8 9     1 2 3   4 5 6   7 8 9
```

Però traslladant la mateixa acció als requadres central o superior serà inevitable sentir un calfred quan, en activar la 13ª casella (representacions del mig i de la dreta, on les caselles prèviament emplenades apareixen en negreta) el caseller 3×3 torni a caure en estat catalèptic. De moment, us n'oferim les taules incompletes i després tractarem d'esbrinar les causes del col·lapse.

Exploració i emplenament de les caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà **2.214** + **323** + **351** = **2.888** retrocessos
- Si també omplim 2,4 (**1**) hi haurà **1.968** + **261** + **114** = **2.343** retrocessos
- Si també omplim 3,4 (**4**) hi haurà **1.513** + **200** + **47** = **1.760** retrocessos
- Si també omplim 4,4 (**3**) hi haurà **1.414** + **128** + **32** = **1.574** retrocessos
- Si també omplim 5,4 (**6**) hi haurà **830** + **120** + **30** = **980** retrocessos
- Si també omplim 6,4 (**5**) hi haurà **638** + **94** + **18** = **750** retrocessos
- Si també omplim 7,4 (**8**) hi haurà **552** + **89** + **9** = **650** retrocessos
- Si també omplim 8,4 (**9**) hi haurà **369** + **58** + **6** = **433** retrocessos
  i també s'omple 9,4 (**7**)
- Si també omplim 1,5 (**3**) hi haurà **1.104** + **173** + **80** = **1.357** retrocessos
- Si també omplim 2,5 (**6**) hi haurà **920** + **137** + **23** = **1.080** retrocessos
- Si també omplim 3,5 (**5**) hi haurà **737** + **108** + **37** = **882** retrocessos
- Si també omplim 4,5 (**?**) hi haurà **?** + **?** + **?** = **?** retrocessos

Exploració i emplenament de les caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà **2.214** + **310** + **215** = **2.739** retrocessos
- Si també omplim 2,7 (**3**) hi haurà **1.968** + **208** + **122** = **2.298** retrocessos
- Si també omplim 3,7 (**1**) hi haurà **1.513** + **414** + **41** = **1.968** retrocessos
- Si també omplim 4,7 (**6**) hi haurà **1.414** + **202** + **16** = **1.632** retrocessos
- Si també omplim 5,7 (**4**) hi haurà **1.064** + **180** + **16** = **1.260** retrocessos
- Si també omplim 6,7 (**2**) hi haurà **821** + **137** + **8** = **966** retrocessos
- Si també omplim 7,7 (**9**) hi haurà **669** + **61** + **9** = **739** retrocessos
- Si també omplim 8,7 (**7**) hi haurà **446** + **40** + **6** = **492** retrocessos
  i també s'omple 9,7 (**8**)
- Si també omplim 1,8 (**6**) hi haurà **1.338** + **145** + **44** = **1.527** retrocessos
- Si també omplim 2,8 (**4**) hi haurà **1.087** + **107** + **22** = **1.216** retrocessos
- Si també omplim 3,8 (**2**) hi haurà **893** + **116** + **14** = **1.023** retrocessos
- Si també omplim 4,8 (**?**) hi haurà **?** + **?** + **?** = **?** retrocessos

En la pàgina 183, després d'una de les plantades de **RE-MEMBER**, insinuàvem que les esperes llargues tant podien acabar en veredicte de compatibilitat com no: "... se'ns escapa de les mans, fins el punt d'haver-nos de preguntar si n'hi ha o no, de solució: n'hi hauria prou a imaginar una cerca ben accidentada i plagada de ziga-zagues, en què s'arribés a l'última casella lliure però no pas per assignar-li l'únic valor possible sinó per trobar-se que ja no en quedava cap", i ara hi haurà ocasió de veure'n una (o dues) d'aquestes, tot i que la inacabable espera no és deguda al fet que l'exploració quedi bloquejada precisament en l'última casella lliure sinó a que, impedits d'anar endavant, hem de remuntar-la fins la primera. La causa d'aquest desgavell és la mateixa en ambdós casos: en la representació del mig, en el primer tercet de la 6ª fila ha d'haver-hi un **7**, un **8** i un **9**, i com que en el tercer tercet de la 4ª fila hi figuren aquests mateixos valors, també hauran de sortir en el segon tercet de la 5ª fila, i per tant el candidat **1** no serà vàlid per a la casella 4,5; en la representació de la dreta, en el primer tercet de la 9ª fila ha d'haver-hi un **7**, un **8** i un **9**, i com que en el tercer tercet de la 7ª fila hi figuren aquests mateixos valors, també hauran de sortir en el segon tercet de la 8ª fila, i per tant el candidat **1** no és vàlid per a la casella 4,8. La forma

en què **RE-MEMBER**, que actua cegament i desconeix aquestes circumstàncies, acabarà descobrint-ho és similar en ambdós casos però encara més rocambolesca en el segon: li costarà molt poc permutar els valors **1** i **2** (subratllats en la representació) i, després del candidat **1** (entre parèntesis per indicar que no és viable), provarà en un moment tots els binomis possibles (**2-4**, **2-7**, **2-8**, **4-2**, **4-7**, **4-8**, **7-2**, **7-4**, **7-8**, **9-2**, **9-4**, **9-7** i **9-8** en el primer cas, i només **7-8** en el segon, sent aquests últims valors els que figuren a les representacions) sense poder passar d'aquí perquè la casella següent (marcada amb X) ja no admet cap valor, i no tindrà més remei que començar a retrocedir fins que el primer requadre 9×3 s'hagi reconvertit a patró NATURMAX adaptat a les preexistències; però en el segon cas, abans d'arribar al tercer requadre i quedar bloquejat a la X que hi hem marcat, se'n haurà trobat una altra a la casella 4,6, que l'obligarà a realitzar una reculada considerable fins el primer requadre 9×3. Hi ha una forma senzilla d'esbrinar si, pel motiu apuntat, és de debò molt més llarga l'exploració del segon cas que la del primer: emplenar les 13 primeres caselles del primer requadre abans de passar al segon o al tercer. Així, de passada, veuríem si amb aquesta ocupació (per sota de la qual no caldria preocupar-se perquè el dispositiu d'ordenació de requadres 9×3 ja intervindria) en tenim prou per fer baixar l'espera fins un punt raonable o bé cal anar pensant en un corrector més. Però, per si de cas no quedés prou clar què pretenem (tot i que no és el primer cop que ho posem en pràctica), convindria fer un incís.

Acceptant que amb una reordenació que traslladi el requadre conflictiu a la part inferior de l'escaquer se solventaran tots els problemes de durada excessiva de les exploracions **RE-MEMBER**, ens podem estalviar mecanismes específics, confiant el control de durades al dispositiu reordenador, si alguna d'aquestes dues sentències és certa:
- L'estat d'emplenament del primer requadre mai no inhibirà la reordenació.
- L'estat d'emplenament del primer requadre pot inhibir la reordenació, però quan això passi la durada del procés també disminuirà, assolint un valor acceptable, per l'augment global d'ocupació.

L'experiència, com acostuma a passar, ha servit per resoldre la qüestió secundària (efectivament el temps d'espera és menor en el primer cas que en el segon) però no la principal: si en un cas l'espera ha caigut a uns pocs segons, a l'altre no s'ha escurçat satisfactòriament i tampoc hem tingut la paciència de romandre asseguts fins que aparegués informació en pantalla. A títol de curiositat, veieu a la dreta la representació del primer cas (limitada als dos primers requadres 9×3) i a l'esquerra la quantitat de retrocessos en l'emplenament 26è (casella 4,5),

```
     1< 2.521 +  8.640 + 0 = 11.161
     2< 2.521 +  8.640 + 0 = 11.161
     8>    60 +     28 + 3 =     91
     9>    60 +     26 + 3 =     89
     T> 5.162 + 17.334 + 6 = 22.502
```

on convé observar que el refús dels candidats **1** i **2** admesos en primera instància, s'indica amb 1< i 2<.

```
0 0 0   0 0 0   0 0 0
3 6 5  (1)0 0   0 0 0
2 1 4   3 6 5   8 9 7
0 0 0   0 0 0   0 0 0
4 5 6   7 0 0   0 0 0
1 2 3   4 5 6   7 8 9
```

A l'hora d'incorporar en el complex **MASCARA** - **RE-MEMBER** un nou dispositiu *ad hoc* per salvar l'escull, és inevitable la reflexió següent: quan llegíeu que, en el primer [segon] cas, a la 6ª [9ª] fila havia d'haver-hi un **7**, un **8** i un **9**, i com que en el tercer tercet de la 4ª [7ª] fila hi figuràven aquests mateixos valors, també havíen de sortir en el segon tercet de la 5ª [8ª] fila, ¿tot plegat no us sonava a *déjà vu*?; i, si parlem del **supòsit 3**, tampoc us sonarà? És clar que sí, i potser llavors us vindrà a la memòria la segona formulació dels 6 supòsits, una generalització que es basava en l'anàlisi de les pseudomatrius **A-9*9** i, sobretot, **A-LLL**, i que es concretava en la supervisió del compliment d'algunes de 3 normes. Doncs, ¡ves per on, que ja donàvem aquesta eina per inútil i encara resultarà que podrem treure'n algun profit!

Igual que llavors examinàvem les **A-9*9** i **A-LLL** d'uns estats d'emplenament concrets (que inicialment s'incrivien en sessions d'exploració un pas per endavant a càrrec d'**EXPL-SUBCONJ** < **EXPLORA-3*9** < **EXPLORA+1** < **ACTUALITZA**, després d'haver ocupat una nova casella, per veure si a partir d'aquell moment calia confiar a **RE-MEMBER** la selecció dels candidats, i després se situaven com una instància prèvia a aquesta funció en **EXPL-SUBCONJ** < **EXPLORA-3*9** < **MASCARA**), ara hauríem d'examinar les **R-9*9** i **R-LLL** corresponents a les proves d'aptitud del candidat **1** en les caselles 4,5 (segona de les representacions a l'inici de la pàgina precedent) i 4,8 (tercera),

a càrrec de **RE-MEMBER**. No cal progresar en l'emplenament virtual de les caselles lliures seguint el patró NATURMÍN, ni tampoc haurem de repetir l'exploració 3 cops (els 9 requadres 3×3, les 9 files i les 9 columnes): n'hi haurà prou acceptant la presència virtual del candidat 1 (**INI** = **T**) i, de moment, explorar els 9 requadres. Si en el primer cas agrupem els elements de **R-9*9** (esquerra) i de **R-LLL** (dreta) que corresponen al requadre 3×3 mitjà dret (centrat a la casella 8,5), tindrem:

```
(6 5) (7 5) (8 5) | (1 2 3 4 5 6 0 0 0) (1 2 3 4 5 6 0 0 0) (1 2 3 4 5 6 0 0 0)
                  |
(6 4) (7 4) (8 4) | (0 2 0 4 0 0 0 0 0) (0 2 0 4 0 0 0 0 0) (0 2 0 4 0 0 0 0 0)
                  |
  8     9     7   |          0                   0                   0
```

Com abans, per tal d'estalviar espai representem amb 0 els valors **nil**. Com que la **norma 1** es respecta, ens centrarem en la dreta, on la 2ª fila mostra 3 llistes que només contenen 2 elements no nuls (**2** i **4**), infringint la **norma 2**, i la 1ª fila mostra com un subconjunt de 4 elements (**1**, **3**, **5** i **6**) només és present a 3 llistes, infringint la **norma 3**.

Si en el segon cas agrupem els elements de **R-9*9** (esquerra) i de **R-LLL** (dreta) que corresponen al requadre 3×3 superior dret (centrat a la casella 8,8), tindrem:

```
(6 8) (7 8) (8 8) | (1 2 3 4 5 6 0 0 0) (1 2 3 4 5 6 0 0 0) (1 2 3 4 5 6 0 0 0)
                  |
(6 7) (7 7) (8 7) | (0 0 3 0 5 0 0 0 0) (0 0 3 0 5 0 0 0 0) (0 0 3 0 5 0 0 0 0)
                  |
  9     7     8   |          0                   0                   0
```

També ens centrarem en la dreta, on la 2ª fila mostra 3 llistes que només contenen 2 elements no nuls (**3** i **5**), infringint la **norma 2**, i en la 1ª fila un subconjunt de 4 elements (**1**, **2**, **4** i **6**) només és present a 3 llistes, infringint la **norma 3**.

Si, enlluernats per la satisfacció de recuperar un procediment d'anàlisi que havia costat mans i mànigues i del qual finalment havíem hagut de prescindir, diguéssim que havíem tingut a tocar la solució definitiva, ens estaríem penjant una medalla immerescuda perquè, deixant de banda la sort d'haver pogut reciclar **EXPLORA-3*9** (que, si seguim utilitzant en versió restringida a requadres 3×3, potser hauríem d'anomenar **EXPLORA-9**), l'estratègia que defensàvem no podia estar més allunyada del que sostenim ara: preteníem que, des de fora de **RE-MEMBER**, es podien expedir certificats d'aptitud fiables per als candidats i detectar el moment a partir del qual només **RE-MEMBER** era solvent; ara, en canvi, sabem que **RE-MEMBER** és l'única que pot abordar aquesta feina i el que fem és posar **ACTUALITZA-PLUS** i **EXPLORA-3*9** al seu servei per tal d'estalviar-li recorreguts innecessaris.

EXPLORA-3*9 recuperarà l'estructura que tenia quan era reclamada des d'**EXPLORA+1** < < **ACTUALITZA**, prèvia a la versió en què feia tàndem amb **RE-MEMBER** des de **MASCARA**. En aquest sentit perd els arguments **A-9*9**, **A-LLL**, **A-N** i **J**, i la part introductòria

```
(setq POST () A-9*9+1 () E -1)
(foreach I A-9*9
  (setq E (1+ E)
        F (if (= E (cadr J)) (subst A-N J I) I)
        A-9*9+1 (cons F A-9*9+1)))
(setq A-9*9+1 (reverse A-9*9+1)
      A-LLL+1 (ACT-LLL (car J) (cadr J) A-LLL))
```

que ja no necessita perquè disposarà d'unes pseudomatrius **R-9*9** i **R-LLL** que venen actualitzades de **RE-MEMBER**, i n'eliminarem les expressions de la segona meitat

```
(setq X ())
(repeat 2
  (if (not POST)
      (progn
        (if X (setq A-9*9+1 (TRANSPOSAR A-9*9+1) A-LLL+1 (TRANSPOSAR A-LLL+1)))
        (setq X T Y -1)
        (repeat 9
          (if (not POST) (progn ...))
          (if (not POST) (progn ... (EXPL-SUBCONJ ()))))))))
```

corresponents a les exploracions per files i per columnes. Introduint la variable local **FORA**, que substituirà **POST** en la pròpia funció i a **EXPL-SUBCONJ**, i canviant de nom altres variables, **EXPLORA-3*9** haurà quedat reduïda a:

```
(defun EXPLORA-3*9 (/ FORA X Y E F R-9 R-L L LL J M)
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y R-9*9 (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y R-LLL (list X Y) (list (+ X 2) (+ Y 2))) L ())
;           ************************************************************************
            (foreach F R-9                                                          ;*
              (foreach E F (if (atom E) (setq L (cons E L)))))                      ;*
            (foreach F R-L                                                          ;*
              (foreach E F                                                          ;*
                (if E (foreach M E                                                  ;*
                        (if (and M (not (member M L))) (setq L (cons M L)))))))     ;*
            (if (< (length L) 9) (setq FORA T))                                     ;*
;           ************************************************************************
            (if (not FORA)
                (progn
                  (setq L ())
                  (foreach F R-L
                    (foreach E F
                      (if E (progn
                              (setq J ())
                              (foreach M E (if M (setq J (cons M J))))
                              (setq L (cons J L))))))
                  (EXPL-SUBCONJ ()))))))
  FORA)
```

En què, per seguir emmarcant amb asteriscs el codi de verificació de la norma 1, no l'hem simplificada fent ... **(if (< (length L) 9) (setq FORA T) (progn ...))**... I, pel que fa a **RE-MEMBER**, només haurà calgut modificar les parts subratllades:

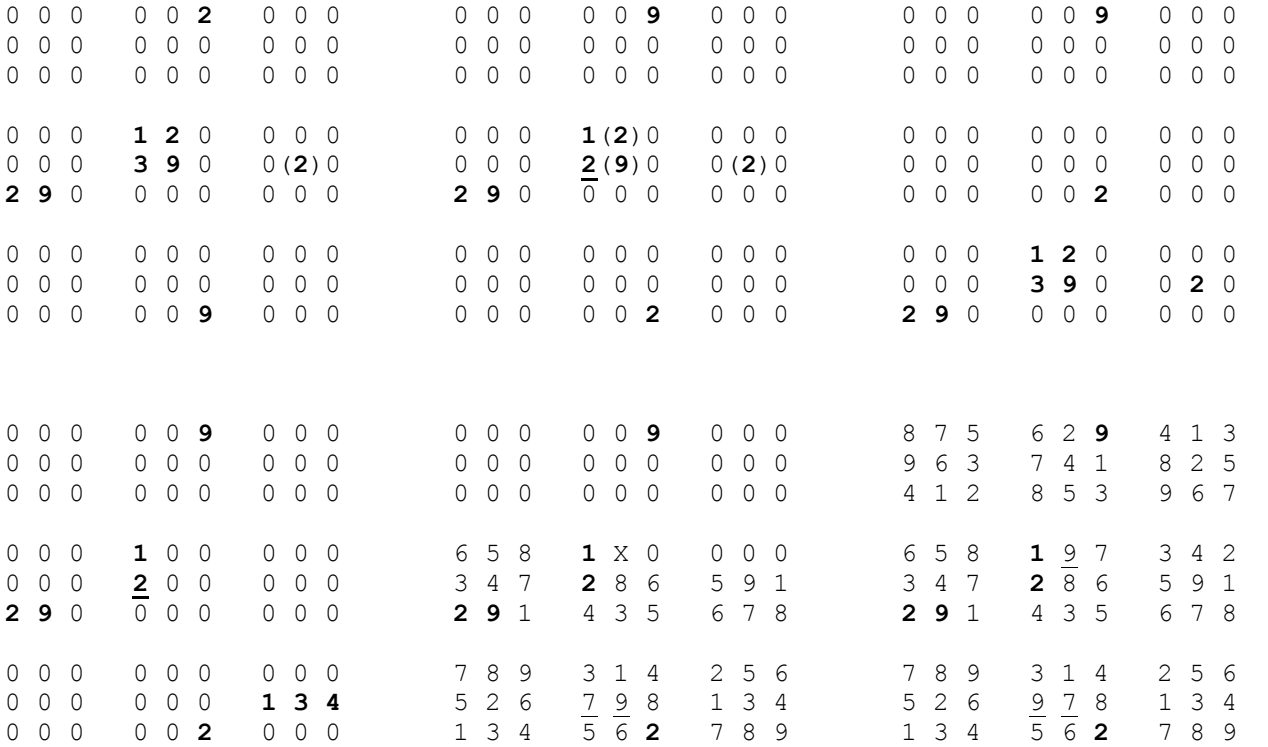
```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL T)
                R-9*9 (car 2R) R-LLL (cadr 2R) INI (EXPLORA-3*9)))
  (if (not (or INI OK)) ; equival a (and (not INI) (not OK))
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
;           (foreach E (reverse L1A9) ; per a NATURMAX
            (foreach E L1A9           ; per a NATURMIN
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

D'una banda es presenta el dubte de si ens curem prou en salut limitant la cerca als requadres 3×3, sense perdre més temps a fer-ho per files i per columnes, i de l'altra una qüestió que probablement el lector ja s'havia preguntat a propòsit de la incorporació d'**ACTUALITZA-PLUS** (quan **ACTUALITZA** és reclamada des de **RE-MEMBER**) i de la més recent recuperació d'**EXPLORA-3*9**: ¿per què limitar-les a l'estat **INI** (és a dir, a les situacions definides per l'ocupació efectiva de caselles i per cada candidat a la casella actual) en lloc d'estendre-la a totes les recursions? Si la seguretat no tingués cost, no ens molestaríem a indagar si tanta cobertura és imprescindible: l'adoptaríem, i prou. Però no volem actuar gratuïtament, perquè la durada de cada prospecció **RE-MEMBER** i del conjunt en cada casella (**MASCARA**) ja freguen el límit tolerable (tant és així que quan **EXPLORA-3*9** s'hagi d'estendre a tot arreu no gosarem aplicar-lo a cada casella i la limitarem als canvis de fila).

Tot i que no ho acostumem a fer (i potser convindria), interrompem la linealitat de l'exposició per avançar els resultats d'aquestes indagacions:

- De les dues etapes en què podem considerar dividida l'acció d'**EXPLORA-3*9**, només la primera (supervisió de la **norma 1**, que no suposa un cost excessiu) s'estendrà a tot l'emplenament virtual. La segona (supervisió de la **norma 2** o **3**, força més onerosa) es limitarà als valors candidats a la casella actual (**INI** = **T**).
- **ACTUALITZA-PLUS** s'estendrà tot l'emplenament virtual, perquè la seva eficàcia en la detecció precoç de camins que no menen enlloc és superior a la d'**EXPLORA-3*9** (**norma 2**/**3**) i el temps que hi esmerça és assumible, malgrat que de bon principi havíem sostingut el contrari muntant un paràmetre *ad hoc* (**REAL**, que se seguirà utilitzant però amb un altre propòsit) per marcar els accessos a **ACTUALITZA.**
- Pel que fa a multiplicar per tres l'acció d'**EXPLORA-3*9** (reduït definitivament a la supervisió de la **norma 1**), efectuant exploracions per files i columnes a més de fer-ho per requadres 3×3, ho deixarem córrer, perquè no hem trobat cap retard imputable a aquesta carència en cap dels casos assajats.
- Com que no suposa cap dispendi apreciable de temps, es manté la implementació del dispositiu previst per aconseguir que durant tot el procés d'emplenament el sudoku estigui ordenat de més a menys ocupació, amb una doble intenció: ajudar, ni que sigui de forma modesta, a escurçar uns temps ordinaris que encara pequen de llargs, i com mesura addicional per esmorteir l'impacte d'alguna eventualitat extraordinària que pogués esdevenir-se malgrat totes les cauteles adoptades.

Dos capítols enrera vam fer la precisió següent: "de fet,únicament s'han detectat situacions en què la prospecció per requadres no ha revelat cap anomalia però sí que ha saltat l'alarma en fer-la posteriorment per files o per columnes, si just abans havia actuat **ACTUALITZA-PLUS** omplint forats". Disortadament, l'autor no ha sabut localitzar els exemples on havia fet aquesta constatació (de vegades, els descobriments se succeeixen tan precipitadament que costa de trobar la parsimònia necessària per ordenar i etiquetar els papers que s'amunteguen a la taula), i va ser precisament quan tractava de reproduir les condicions esmentades (que, després d'un emplenament automàtic, l'escaquer quedés just a punt de vulnerar algun dels 6 supòsits) per veure si en el nou context també era indispensable el triple filtre, que es va produir un inesperat col·lapse. Aquesta incidència, que tot seguit us descriurem, no ha dut més claror sobre aquell particular però sí que ha servit per constatar la necessitat d'ampliar el control de compliment de les 3 normes (si més no, de la **norma 1**) a totes les recursions de **RE-MEMBER.**



La representació de dalt, a l'esquerra, mostra una prova on després d'emplenar les caselles 1,4 (**2**), 2,4 (**9**), 6,1 (**9**), 6,9 (**2**), 4,6 (**1**) i 4,5 (**3**), en aquest ordre, es comprovava que si omplíem amb un **2** [**9**] alguna de les dues caselles 5,6 i 5,5, **ACTUALITZA-PLUS** ens col·locava un **9** [**2**] a l'altra. La intenció era complicar una mica més aquest joc, de manera que l'assignació del **2** [**9**] no fos manual sinó també automàtica, conseqüència d'haver emplenat prèviament 8,5 o 8,6 amb un **2** [**9**]. Però

en tornar a repetir l'operació per introduir aquesta complicació i en experiències successives anar-ho complicant més i més, una distracció accidental (que en teoria no havia de canviar res) va fer que en 6,1 hi anés el **2** i a 6,9 el **9**. Res de nou en emplenar 4,6 (**1**), però en activar 4,5 va sobrevenir la suspensió aparent de tot signe d'activitat, com veieu al mig, on teniu el candidat **2** (en la verificació del qual ha hagut de produir-se el col·lapse, perquè en pantalla no hi surt cap traça) i les frustrades asignacions posteriors es representen entre parèntesis. Abans de res, s'ha provat què passaria permutant els dos primers requadres 9×3, com faria la rutina que tenim en cartera: es representa a la dreta, ja amb el valor previst a 4,5 (ara 4,2) i els restants sense parèntesis perquè no hi ha hagut cap incident i s'ha pogut completar la seqüència. Però una reflexió que hem fet altres vegades, materialitzada a baix, a l'esquerra, ens obliga a anar més enllà: la permutació de requadres no es produiria en un cas com aquest, per la presència dels valors **1**, **3** i **4** que, amb el **2**, converteixen el requadre inferior en el més poblat.

Aquesta vegada gairebé no cal dependre de la informació que ens podria facilitar la funció **RE-MEMBER** oportunament preparada, perquè el procés d'emplenament virtual sota la presència del candidat **2** a 4,5 és fàcil reproduir-lo manualment, en haver-hi molts pocs retrocessos fins arribar a la posició 5,6, on ja no queden candidats i on començarà un llarg i penós retrocés fins la casella 4,2, que és la causa de la frenada. A baix, al mig, podeu seguir aquest emplenament i veure que la causa de tot és la presència d'un **9** a la casella 5,2, valor que bloqueja el recuadre 3×3 central: les dues úniques posicions d'aquest requadre que admetien un **9**, a partir dels emplenaments reals (caràcters en negreta, inclòs el de la casella actual), eren 5,5 i 5,6, i aquesta possibilitat ha quedat exclosa des que **RE-MEMBER** ha fet l'assignació virtual. Repetim un cop més: nosaltres ho veiem però **RE-MEMBER** no, i l'única opció que té és retrocedir des de 5,6 fins 4,2 (no és un retrocés lineal sinó zigzaguejant, com el que abans havíem caricaturitzat parlant de 10 passos enrera i 9 endavant) on, després d'haver permutat els valors **7** i **9** subratllats, ja només queda avançar de nou pel camí registrat a la dreta, aquesta vegada gairebé sense entrebancs i directe al final. Ara bé, si en **RE-MEMBER** haguéssim ampliat la cobertura d'**EXPLORA-3*9** a tots els recorreguts, més enllà del pas inicial **INI**, no hauria calgut esperar a activar la casella 5,6 sinó que, amb la 5,2 virtualment ocupada per un **9**, el bloqueig que s'havia de produir quatre caselles més amunt i que ja estava cantat hagués estat anunciat per l'esmentada funció, sols veient els elements de **R-9*9** (esquerra) i **R-LLL** (dreta) corresponents al requadre 3×3 central (centrat a la casella 5,5):

```
     1    (5 6) (6 6) |          0          (0 0 3 4 5 0 7 8 0) (0 0 3 4 5 6 7 8 0)
                      |
     2    (5 5) (6 5) |          0          (0 0 3 4 5 0 7 8 0) (0 0 3 4 5 6 7 8 0)
                      |
   (4 4) (5 4) (6 4) | (0 0 3 4 0 6 0 8 0) (0 0 3 4 5 0 7 8 0) (0 0 3 4 5 6 7 8 0)
```

Prenent nota dels valors 1... 9 presents a l'esquerra i a les llistes de la dreta, hi trobem a faltar el **9**, cosa que implica infracció de la **norma 1** (subsidiàriament també s'infringeix la **norma 2**, perquè les 7 llistes de la dreta només contenen 6 elements: **3**, **4**, **5**, **6**, **7** i **8**). Fixeu-vos, a més, que en el cas present no hi ha cap forat per tapar, de manera que **ACTUALITZA-PLUS** no hauria salvat la situació encara que hagués pogut ficar-hi cullerada: sols la intervenció d'**EXPLORA-3*9**, ni que fos sense la 2ª part, només per supervisar la **norma 1**, podria copsar-ne la vulneració; i encara ens haurà anat bé haver articulat **EXPLORA-3*9** de forma que, en comptes de limitar-se a la verificació de la **norma 2** (que engloba l'anterior però és força més llarga), de primer comprovés la **norma 1** i, només si aquesta es complia, passés a comprovar la **norma 2** o **3**. Tot seguit veureu que, amb pocs canvis a **EXPLORA-3*9** i **RE-MEMBER** (el codi addicional hi figura subratllat), es recull aquesta ampliació.

```
(defun EXPLORA-3*9 (/ FORA X Y E F R-9 R-L L LL J M)
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y (car 2R) (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y (cadr 2R) (list X Y) (list (+ X 2) (+ Y 2))) L ())
;           ***************************************************************
            (foreach F R-9                                                 ;*
              (foreach E F (if (atom E) (setq L (cons E L)))))             ;*
```

```
            (foreach F R-L                                                    ;*
              (foreach E F                                                    ;*
                (if E (foreach M E                                            ;*
                        (if (and M (not (member M L))) (setq L (cons M L))))))) ;*
            (if (< (length L) 9) (setq FORA T))                               ;*
;           ********************************************************************
            (if (and INI (not FORA))
                (progn
                  (setq L ())
                  (foreach F R-L
                    (foreach E F
                      (if E (progn
                              (setq J ())
                              (foreach M E (if M (setq J (cons M J))))
                              (setq L (cons J L))))))
                  (EXPL-SUBCONJ ()))))
  FORA)

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL T)
                INI (EXPLORA-3*9)
                R-9*9 (car 2R) R-LLL (cadr 2R)))
  (if (not (or INI OK))
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
;           (foreach E (reverse L1A9) ; per a NATURMAX
            (foreach E L1A9           ; per a NATURMIN
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if (not OK) (setq OK (EXPLORA-3*9)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

De fet, els primers arguments de **SUB-X*Y** (**R-9*9** i **R-LLL**, en la versió precedent d'**EXPLORA-3*9**) ja els hauríem pogut posar en la forma actual **(car 2R)** i **(cadr 2R)**, i si aleshores la tercera línia de **RE-MEMBER**
era ... **R-9*9 (car 2R) R-LLL (cadr 2R) INI (EXPLORA-3*9)))**
i no ... **INI (EXPLORA-3*9) R-9*9 (car 2R) R-LLL (cadr 2R)))**
com ara, en què després de **2R** l'ordre de les assignacions és indiferent, va ser perquè només hi havia una crida a **EXPLORA-3*9** i aquella era l'opció més simple.

Les provatures, a la recerca d'algun cas en què després d'un emplenament automàtic es produís la vulneració d'alguna de les tres normes i que aquesta vulneració sols es manifestés en files o columnes (però no en requadres 3×3, l'únic àmbit en què fins ara s'ha implementat **EXPLORA-3*9**), han prosseguit però sense aconseguir que el programa registrés cap frenada deguda a la causa expressada. Val a dir que sí que n'hi ha hagut, de frenades, però de procedència diversa i que ens anat bé per acabar de polir **RE-MEMBER**. Aquí en teniu tres escenaris, que tractarem un a un:

1 2 0	0 0 0	**3** 0 0
0 0 0	0 0 0	0 0 0
0 0 **3**	0 **7** 0	0 **1 2**
0 0 0	0 0 **6**	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
6 0 0	0 0 0	

1 2 0	0 0 0	**5** 0 0
0 0 0	0 0 0	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 0 0	0 0 **6**	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
6 5 0	0 0 0	

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 **2**	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 **8** 0
0 0 0	**6 5** 0	0 0 0

Començarem amb el primer d'aquests escenaris: a l'esquerra representem de nou la situació de partença del seguiment comparatiu que durem a terme (els emplenaments previs van en negreta, igual que la casella actual 5,7, on a més se subratlla el candidat **7** que **RE-MEMBER** comprova), a la dreta la solució segons el patró NATURMÍN que ratifica la validesa d'aquest candidat, i al mig l'estat d'emplenament virtual que hem escollit per apreciar els diferents temps que l'esmentada funció necessita per rectificar la primera assignació a la casella 4,5 (**3**, que és inviable i que hem subratllat) i substituir-la per una assignació compatible (**8**, a la dreta, que també hem subratllat), depenent dels recursos movilitzats. En realitat l'etapa més llarga va des d'una 3ª fila que és 7-8-9-1-2-4-5-3-6 i passa a 7-8-9-1-4-2-5-3-6, però hem escollit una seqüència posterior per a no estendre'ns massa.

1 2 0	0 0 0	**3** 0 0	**1 2** 0	0 0 0	**3** 0 0	**1 2** 7	9 6 8	**3** 4 5
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	8 4 5	3 2 1	9 6 7
0 0 **3**	0 **7** 0	0 **1 2**	0 0 **3**	0 **7** 0	0 **1 2**	6 9 **3**	5 **7** 4	8 **1 2**
0 0 0	0 0 **6**	0 0 0	0 0 0	0 0 **6**	0 0 0	9 7 8	4 1 **6**	2 5 3
0 0 0	0 0 0	0 0 0	5 6 2	3 1 4	X 0 0	5 6 2	8 9 3	4 7 1
0 0 0	0 0 0	0 0 0	4 3 1	2 5 7	6 9 8	4 3 1	2 5 7	6 9 8
0 0 0	0 0 0	0 0 0	7 8 9	1 4 2	5 3 6	7 8 9	1 4 2	5 3 6
0 0 0	0 0 0	0 0 0	3 5 6	7 8 9	1 2 4	3 5 6	7 8 9	1 2 4
0 0 0	**6** 0 0	0 0 0	2 1 4	**6** 3 5	7 8 9	2 1 4	**6** 3 5	7 8 9

En primer lloc presentem les giragonses amb la versió actual de **RE-MEMBER**, en què **EXPLORA-3*9** ja s'ha estès a tota la funció però limitada al control de la **norma 1**:

1 2 0	0 0 0	**3** 0 0	**1 2** 0	0 0 0	**3** 0 0	**1 2** 0	0 0 0	**3** 0 0
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 0 **3**	0 **7** 0	0 **1 2**	0 0 **3**	0 **7** 0	0 **1 2**	0 0 **3**	0 **7** 0	0 **1 2**
0 0 0	0 0 **6**	0 0 0	0 0 0	0 0 **6**	0 0 0	0 0 0	0 0 **6**	2 5 3
5 6 2	3 1 4	X 0 0	5 6 2	3 1 8	4 7 X	5 6 2	3 9 1	4 7 X
4 3 1	2 5 7	6 9 8	4 3 1	2 5 7	6 9 8	4 3 1	2 5 7	6 9 8
7 8 9	1 4 2	5 3 6	7 8 9	1 4 2	5 3 6	7 8 9	1 4 2	5 3 6
3 5 6	7 8 9	1 2 4	3 5 6	7 8 9	1 2 4	3 5 6	7 8 9	1 2 4
2 1 4	**6** 3 5	7 8 9	2 1 4	**6** 3 5	7 8 9	2 1 4	**6** 3 5	7 8 9

1 2 0	0 0 0	**3** 0 0	**1 2** 0	0 0 0	**3** 0 0	**1 2** 0	0 0 0	**3** 0 0
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 0 **3**	0 **7** 0	0 **1 2**	6 4 **3**	5 **7** X	0 **1 2**	0 0 **3**	0 **7** 0	0 **1 2**
0 0 0	0 0 **6**	0 0 0	8 9 7	4 1 **6**	2 5 3	0 0 0	0 0 **6**	0 0 0
5 6 2	3 9 4	X 0 0	5 6 2	3 9 8	4 7 1	5 6 2	4 0 0	X 0 0
4 3 1	2 5 7	6 9 8	4 3 1	2 5 7	6 9 8	4 3 1	2 5 7	6 9 8
7 8 9	1 4 2	5 3 6	7 8 9	1 4 2	5 3 6	7 8 9	1 4 2	5 3 6
3 5 6	7 8 9	1 2 4	3 5 6	7 8 9	1 2 4	3 5 6	7 8 9	1 2 4
2 1 4	**6** 3 5	7 8 9	2 1 4	**6** 3 5	7 8 9	2 1 4	**6** 3 5	7 8 9

Amb aquests 6 fotogrames, discontinus però suficients per descriure les peripècies de **RE-MEMBER** provant els valors **3** i **4** a la casella 4,5, fins que el torn del **8** ens deixa seguir, ens podem fer una idea del temps perdut en trobar-hi una assignació vàlida. I més si considerem que això és *peccata minuta* comparant-ho amb l'odissea de la casella 5,3 trencant-se les banyes amb el valor **2** abans de trobar llum verda amb el **4**, que com dèiem abans és infinitament més llarga i per això hem preferit no entrar-hi. ¿Algun sistema d'alarma ens en podria alertar, de la casella 4,5 (o de la 5,3) estant? Doncs sí, deixant que **EXPLORA-3*9** campés lliurement, verificant les tres normes, per tot **RE-MEMBER** (ara es limita a supervisar la **norma 1**, perquè el control de la **norma 2** o **3** queda circumscrita a la situació **INI**), però això és inacceptable pel brutal alentiment que suposaria, i aquesta estimació no és una simple conjectura sinó que l'hem comprovada experimentalment. Val a dir que, si hem pogut afirmar que una **EXPLORA-3*9** dotada de carta blanca faria una detecció precoç de candidats sense futur, és perquè, sabent en quines condicions es produïa la marrada, hem intervingut **RE-MEMBER** perquè la plena operativitat d'**EXPLORA-3*9**

només es donés quan els candidats a la casella 4,5 fossin els valors **3** o **4**, amb les precedents tal i com apareixen en les representacions. Efectivament, així que la casella esmentada s'emplena virtualment amb un **3**, detectem infraccions de les **normes 2** i **3** en el requadre 3×3 superior centre (centrat a la casella 5,8), on observant els elements de **R-LLL** (dreta) hi trobarem: 5 llistes que només contenen 4 elements no nuls (**4**, **5**, **8** i **9**) [si ho preferiu, 6 llistes que només en contenen 5 (**4**, **5**, **6**, **8** i **9**)], infringint la **norma 2**, i un subconjunt de 3 elements (**1**, **2** i **3**) que només són presents a 2 llistes [si ho preferiu, un subconjunt de 4 elements (**1**, **2**, **3** i **6**) que només són presents a 3 llistes], infringint la **norma 3**:

```
(3 8) (4 8) (5 8) | (0 0 0 4 5 0 0 8 9) (0 0 0 0 0 6 0 0 9) (0 0 0 4 0 0 0 8 0)
                  |
(3 7) (4 7) (5 7) | (0 0 0 4 5 0 0 8 9) (1 2 0 0 0 6 0 0 9) (1 0 3 4 0 0 0 8 0)
                  |
(3 6)   7   (5 6) | (0 0 0 4 5 0 0 8 9)          0          (0 0 0 4 0 0 0 8 0)
```

Ara bé, si us hi fixeu encara podríem descobrir una altra cosa: els elements **2** i **3** que només apareixen cada un en una llista i ens fan pensar que, si **ACTUALITZA-PLUS** tingués jurisdicció en la present circumstància, hauria de tapar els forats 5,8 i 6,8 amb aquests valors (virtualment, és a dir condicionat-ho a l'emplenament de 5,7 amb un **7**), i quedaria ben palesa la inviabilitat d'aquesta situació, perquè a la casella 5,6 caldria posar-hi un **1** i això ja no és possible. Però és que si la funció esmentada pogués intervenir, també ho faria en el requadre 3×3 mitjà dret (centrat a la casella 8,5), on a **R-LLL** (dreta) tornen a aparèixer els valors **2** i **3** cadascun en una única llista, tot i que aquí sí que seria possible col·locar-los a 7,6 i 9,6, i després res no s'oposaria a que s'emplenés el forat 9,5 amb un **1**:

```
(6 5) (7 5) (8 5) | (0 2 0 4 0 0 0 0 0) (0 0 0 4 5 0 7 0 0) (1 0 3 0 5 0 7 0 0)
                  |
(6 4) (7 4) (8 4) | (0 0 0 4 0 0 0 0 0) (0 0 0 4 0 0 7 0 0) (1 0 0 0 0 0 7 0 0)
                  |
  6     9     8   |          0                  0                    0
```

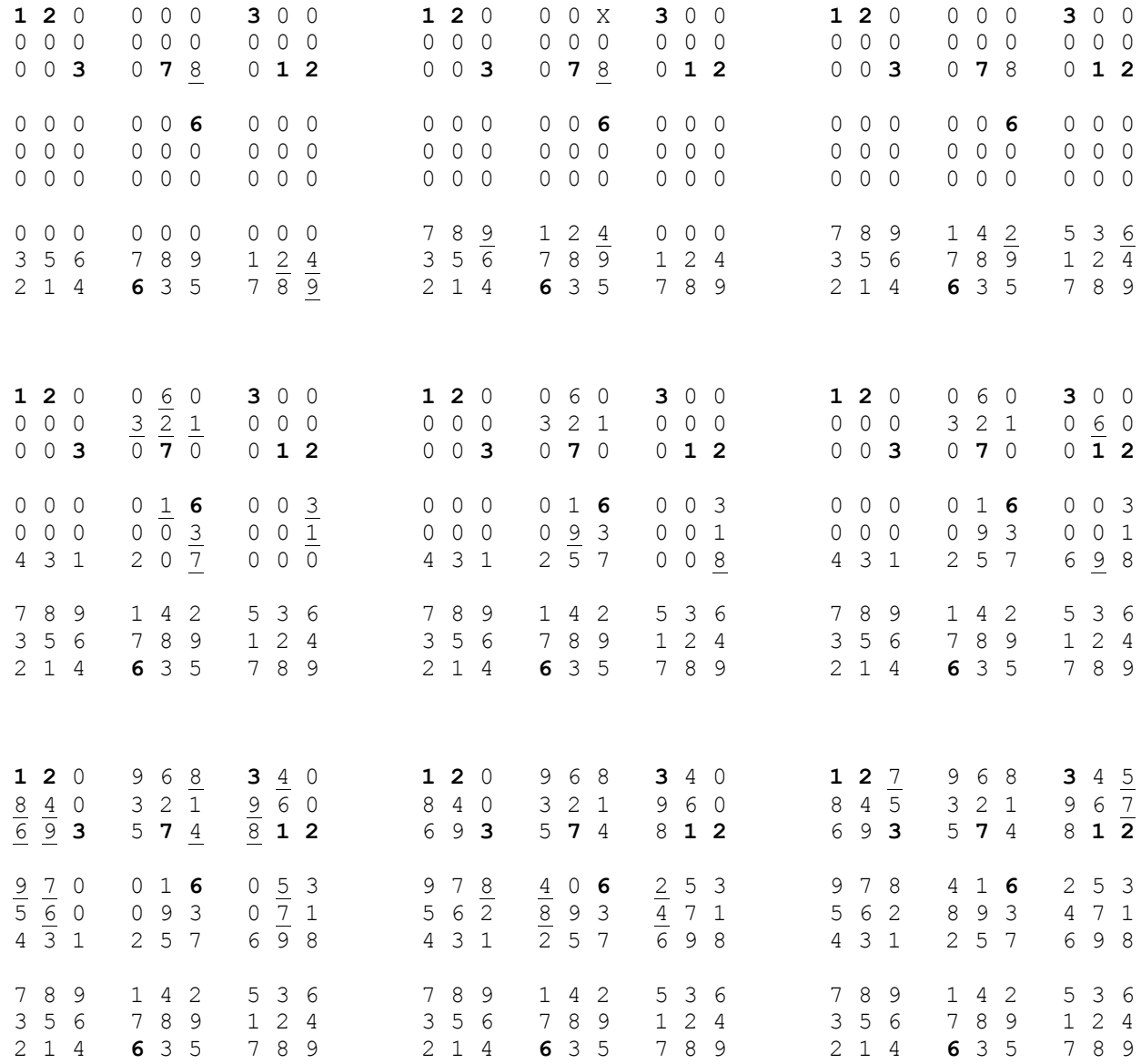
A diferència del candidat **3** a la casella 4,5, que s'enfilava fins la posició 4,7 i que escurçarà el temps de diagnòstic acceptant plenament **EXPLORA-3*9** en **RE-MEMBER**, l'altre candidat inviable, **4**, ja en té prou amb el primer dels dispositiu adoptats per la segona funció, que li permetrà detectar el bloqueig 3 posicions abans de la casella 7,5 (marcada amb una X a l'últim fotograma de la seqüència precedent):

```
(foreach F (cadr 2R)
  (if (and (not OK) (member '(() () () () () () () () ()) F))
      (setq OK T)))
```

Sembla, doncs, que si no hi ha a **R-LLL** cap llista providencial amb tots 9 elements nuls, **EXPLORA-3*9** ens pot advertir amb certa anticipació que ens hen ficat en un carreró sense sortida, independentment de si no és pitjor el remei que la malaltia (sobretot tenint en compte que l'alentiment repercutirà en cada casella a emplenar i del profit només en gaudirem de tant en tant). Però dels costos ja parlarem més endavant, i és per això que, posats a tirar la casa per la finestra, ens atrevim a preguntar-nos si **ACTUALITZA-PLUS**, relegada de moment a **INI**, podria servir-nos per aquesta comesa, i no parlem tant de capacitat per detectar bloquejos a temps sinó per clarificar la situació, reduint sensiblement el nombre d'incògnites (caselles buides) en realitzar emplenaments obvis i, en definitiva, aproximant el moment del veredicte, tant si és negatiu com positiu. En l'exploració per requadres 3×3 que hem fet de **R-LLL** mentre **RE-MEMBER** provava valors a 4,5, ja hi hem trobat llistes que activarien l'acció d'**ACTUALITZA-PLUS** si aquesta funció tingués la possibilitat de participar més enllà de **INI**. Doncs bé, donem-li l'oportunitat d'intervenir però no arrenquem d'una situació ja massa empantanegada sinó veiem què és capac de fer intervenint en cada emplenament de bon principi. En concret, ¿com se'n sortirà del primer emplenament 7-8-9-1-2-4-5-3-6 de la 3ª fila, amb què abans havíem assegurat que s'obria un llarg i tortuós viatge que no duia enlloc, fins albirar un horitzó més clar a 7-8-9-1-4-2-5-3-6?: doncs d'una manera tan senzilla que sembla mentida.

Veiem-ho en les dues primeres representacions del conjunt de 3×3 que teniu girant full, on els emplenaments automàtics estan subratllats. En la primera hi apleguem el toc final de les dues files inferiors: la casella final de la 1ª fila (**9**) i de la 2ª (**4**), amb la penúltima, 8,2 (**2**), de propina. Aquest inesperat automatisme és degut a la presència del **2** a 9,7: així que haguem emplenat 7,2 (**1**), l'assignació a

9,2 (**4**) queda determinada i **ACTUALITZA-PLUS** actua tapant el forat; després d'això, queda determinada l'assignació a 8,2 (**2**) i la funció torna a actuar. No perdrem el temps presentant els elements de **R-LLL** que desencadenen l'acció, perquè en treurem més suc en la jugada següent. En la segona representació hi apleguem el toc final dels dos primers requadres 3×3: l'última casella del primer (**9**) i del segon (**4**), amb la casella 6,7 (**8**) de propina. Aquest inesperat automatisme ara és degut a la presència del **7** a 5,7: així que haguem emplenat 5,3 (**2**), l'assignació a 6,3 (**4**) queda determinada i **ACTUALITZA-PLUS** actua tapant el forat. Després d'això, ¿hem de repetir que l'assignació a 6,7 (**8**) queda determinada i la funció torna a actuar?: doncs, no exactament. Si penseu en quins són els elements de **R-LLL** corresponents a les caselles 6,7 i 6,9, just després que **ACTUALITZA-PLUS** hagi emplenat 6,3 (**4**), us trobareu dues llistes **(nil nil nil nil nil nil nil 8 nil)**. **ACTUALITZA-PLUS** podria emplenar-ne una qualsevol, però com que la cerca es fa ordenadament per files, li tocarà a la primera: l'element de **R-LLL** associat passarà a **nil**, el corresponent a la posició 6,9 es transformarà en **(nil nil nil nil nil nil nil nil nil)** i aquesta llista serà detectada pel dispositiu reproduït a la pàgina precedent, provocant la fi de la progressió, l'anul·lació de l'emplenament 6,3 (**4**) i les seves seqüeles i, com que ja no queden candidats, el retorn a la casella precedent per provar amb **4**.



Les set representacions restants descriuen l'exploració a partir d'aquest valor, que amb l'ajut d'**ACTUALITZA-PLUS** es converteix en una marxa triomfal. Completant la renglera superior, la primera (dreta) recull el toc final del segon i tercer requadres 3×3: 6,3 (**2**) i 9,3 (**6**). Passant a la segona renglera, l'emplenament de la casella 4,4 (**2**) (esquerra) desencadena una seqüència d'emplenaments automàtics, 5,8 (**2**), 6,8 (**1**), 4,8 (**3**), 6,5 (**3**), 6,4 (**7**), 9,6 (**3**), 9,5 (**1**), 5,6 (**1**) i 5,9 (**6**); el de la casella 5,4 (**5**) (centre) desencadena l'emplenament de 9,4 (**8**) i 5,5 (**9**), i el de la 7,4 (**6**) (dreta) desencadena els de 8,4 (**9**) i 8,8 (**6**). Passant finalment a la tercera renglera, l'emplenament de la casella 1,5 (**5**) (esquerra) desencadena

una llarga sèrie d'emplenaments automàtics, 2,5 (**6**), 8,6 (**5**), 8,5 (**7**), 8,9 (**4**), 6,9 (**8**), 6,7 (**4**), 2,7 (**9**), 2,6 (**7**), 4,7 (**5**), 7,7 (**8**), 1,7 (**6**), 1,8 (**8**), 1,6 (**9**), 2,8 (**4**), 7,8 (**9**) i 4,9 (**9**); el de la casella 3,5 (**2**) (centre) en desencadena una altra, 7,5 (**4**), 4,5 (**8**), 3,6 (**8**), 4,6 (**4**) i 7,6 (**2**), i l'emplenament de la 3,8 (**5**) (dreta) desencadena el de 9,8 (**7**), 3,9 (**7**) i 9,9 (**5**).

Cadascuna amb una estratègia específica, tant **EXPLORA-3*9** en règim intensiu (és a dir, no limitat al control de la **norma 1**) com **ACTUALITZA-PLUS** aconsegueixen reduir dràsticament, intervenint en tot l'àmbit de **RE-MEMBER** (el primer accés **INI** i les demés autorecursions), el recorregut per l'escaquer, fins a obtenir el veredicte de compatibilitat dels candidats. Ara bé: passa el mateix amb els temps d'espera? Doncs, rotundament no.

L'extensió d'**EXPLORA-3*9** a tot l'àmbit de **RE-MEMBER**, que s'aconsegueix retornant la primera funció a la penúltima versió, és a dir, substituint la condició d'accés a la supervisió de **norma 2** o **norma 3**, **(if (and INI (not FORA)) ...)** per l'anterior **(if (not FORA) ...)**, l'hauríem de deixar en **(if (and NOU-Y (not FORA)) ...)**, on la variable **NOU-Y**, locals en **MASCARA**, jugarien a **RE-MEMBER** el paper següent:

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL T) NOU-Y T REF-Y -1
                INI (EXPLORA-3*9)
                R-9*9 (car 2R) R-LLL (cadr 2R)))
  (if (not (or INI OK))
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq NOU-Y (/= (cadr R-P) REF-Y)
                  REF-Y (if NOU-Y (cadr R-P) REF-Y)
                  R-N (nth (car R-P) (nth REF-Y R-LLL)))
;           (foreach E (reverse L1A9) ; per a NATURMAX
            (foreach E L1A9           ; per a NATURMIN
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if (not OK) (setq OK (EXPLORA-3*9)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

¿Per què aquesta última complicació, remarcada com de costum amb els subratllats? ¿Que no n'hi havia prou a deixar com abans **EXPLORA-3*9**, per aconseguir que els dos accessos a aquesta funció des de **RE-MEMBER** s'executessin igual, sense restringir a **INI** el control de les **normes 2** o **3**? Sí, però d'aquí venia el problema: l'execució d'**EXPL-SUBCONJ** és tan llarga que, si cada auto-recursió de **RE-MEMBER** (cada avanç de casella) havia de dur-ne una d'aparellada, el degoteig visual de candidats en el caseller 3×3 de **MASCARA** era exasperant. La solució salomònica que hem proposat consisteix a no realitzar la 2ª part de l'exploració **EXPLORA-3*9** cada vegada que emplenem virtualment una casella, sinó només quan el pas endavant comporti també canvi de fila. Més que salomònica n'hauríem de dir descafeïnada, perquè si d'una banda aconseguim contenir la durada resultant (suposant que **RE-MEMBER** actui sobre un escaquer pràcicament buit, l'exploració completa només es farà en una de cada deu caselles visitades) de l'altra en reduïm l'eficiència (la detecció precoç de configuracions condemnades no ho serà tant, de precoç, o simplement no arribarà a produir-se i, com abans, descobrirem sobtadament que la següent casella lliure ja no té cap valor assignable en primera instància). En resum: l'intèrval mínim d'aparició de candidats en el caseller 3×3 serà més llarg del que fóra desitjable i, dintre de les esperes que superin aquest mínim, només aconseguirem escapçar les de mitja i llarga durada, perquè les curtes se seguiran notant.

Paradoxalment i per a vergonya de l'autor (que sembla no haver escarmentat prou amb els prejudicis que havia manifestat en contra de l'ús extensiu de **RE-MEMBER**, prejudicis que es van encarregar de desmentir unes experiències que hagués calgut fer de bon principi), el tan temut **ACTUALITZA-PLUS** no era tan lent com li semblava quan dos capítols enrera, en presentar **RE-MEMBER**, havia deixat escrit:

"Tal i com ha quedat **ACTUALITZA**, el tapaforats **ACTUALITZA-PLUS**, d'ús obligat quan **POST** = **nil** (per evitar sudokus sense solució) i optatiu quan **POST** = **T** (per mirar de sostreure a **RE-MEMBER** algunes caselles i accelerar el procés), s'aplica sempre però només sobre els valors realment assignats a les caselles (**REAL** = **T**), perquè per fer-ho també quan **RE-MEMBER** busca la viabilitat d'aquests valors (**REAL** = **nil**) no seria rendible: introduir **ACTUALITZA-PLUS** a cada recursió encara alentiria el procés."

És cert que, quan afirmàvem això, **ACTUALITZA-PLUS** encara era una farragosa bateria de plantilles per detectar diverses situacions particulars definides en **R-9*9**, i que més tard, quan vam aconseguir una formulació genérica definida en **R-LLL** que englobés aquestes situacions i reduís el temps de processat, no ens vam molestar a revisar una estimació precipitada que probablement era falsa des del començament. Ja és el segon cop que l'autor se l'ha d'embeinar fent-se enrera d'una asserció imprudent que no havia contrastat amb la pràctica, però no li sap greu admetre-ho perquè, al cap i a la fi, ha estat possible avançar gràcies al fet que les seves pressumpcions eren errònies. El més curiós de tot és que, posats a acceptar que **EXPLORA-3*9** o **ACTUALITZA-PLUS** (l'una o l'altra, perquè admetre les dues suposaria fregar uns límits de paciència que només el Job bíblic podria superar) treguin el nas per **RE-MEMBER** (en totes les recursions, i no només quan **INI** = **T**), l'afectació de la segona és lleugerament menor que la de la primera. Si tot fos una qüestió de grau, encara ens veuríem obligats a aprofundir el tema per tal de veure si aquesta ponderació desigual és privativa de unes circumstàncies concretes que, si no es donen, poden invertir la correlació. Sortosament per a nosaltres no caldrà filar tan prim, perquè a través dels exemples que hem anat improvisant (sempre amb la mala intenció de provocar fallades cada cop més sonades) ha quedat prou palès que, on **EXPLORA-3*9** pugui salvar la situació, també ho farà **ACTUALITZA-PLUS**, però sense que la recíproca sigui certa. Així que, amb totes les reserves que calgui adoptar en relació a un discurs inductiu menys que imperfecte, la conclusió provisional és que n'hi haurà prou a estendre a **RE-MEMBER** l'acció d'**ACTUALITZA-PLUS**.

Tanmateix no avancem esdeveniments, i prenem-nos el temps necessari per veure els escenaris anunciats al final de la pàgina 199 i apreciar quin repertori tan divers de situacions poden coincidir en la temuda eclosió d'un temps d'espera inusualment llarg en relació al de la majoria de candidats a rebre l'aprovació de **RE-MEMBER**: així com el segon seguirà la línia del primer, que acabem d'analitzar, quant a la intercanviabilitat d'**EXPLORA-3*9** i **ACTUALITZA-PLUS**, el tercer posarà de manifest la incapacitat de la primera d'aquestes funcions perquè l'itinerari d'exploració s'escurci prou, si més no per rebaixar el temps d'espera fins un límit tolerable; de nou, la típica incursió en fals fins alguna posició avançada, seguida per una penosa retirada zigzaguejant fins gairebé la línia de sortida, alfa i omega d'un peculiar viacrucis sincrètic que podríem imaginar, per no limitar-nos com sempre a referències bíbliques, prenent materials suggerents de les mitologies helènica i judeocristiana per tal de barrejar la pujada al Gòlgota de Jesús portant la creu amb la de Sísif carregant aquella pedra que reiteradament rodola pendent avall poc abans de fer el cim.

Faltava un petit detall: les modificacions sobre els codi precedent per adaptar-hi aquest dispositiu. De primer, una cosa que estranyarà és que la funció **PLUS-N->P** retorni a la versió primitiva:

```
(defun PLUS-N->P ()
  (setq A-N N A-P K PLUS T)
  (if REAL (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))))
```

Per entendre aquesta decisió, caldrà veure com han quedat **ACTUALITZA** i **RE-MEMBER**,

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y LL PLUS E F L)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if REAL (ACTUALITZA-PLUS))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))
```

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                INI (EXPLORA-3*9) R-9*9 (car 2R) R-LLL (cadr 2R)))
  (if (not (or INI OK))
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
;           (foreach E (reverse L1A9) ; per a NATURMAX
            (foreach E L1A9           ; per a NATURMIN
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if (not OK) (setq OK (EXPLORA-3*9)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

i desfer l'últim canvi d'**EXPLORA-3*9**, tornant la condició d'accés a la supervisió de **norma 2** o **norma 3** a la forma **(if (and INI (not FORA)) ...)**. Així, l'únic accés a **ACTUALITZA-PLUS** amb l'argument **REAL** en **T** es farà des d'**OMPLE-SUDOKU**, sistema més econòmic que treure'ns de la màniga una nova variable exclusivament per controlar la visió en pantalla de les assignacions automàtiques d'**ACTUALITZA-PLUS**: com que la funció és accedida tant des de l'àmbit real (directament, des d'**OMPLE-SUDOKU**) com virtual (prospeccions des de **RE-MEMBER** < **MASCARA** < **OMPLE-SUDOKU**), resulta més pràctic que **REAL** deixi de condicionar una acció que ja s'executa sempre i passi a condicionar la visualització, que només s'ha de produir en el primer cas, treient-la d'**ACTUALITZA** i incorporant-la a **PLUS-N->P** < **ACTUALITZA-PLUS**.

Seguim amb el segon dels escenaris escollits, que parteix de la substitució en el primer dels dos valors **3** per **5**, donant la casella **7** per consolidada, considerant que la casella actual és 5,1 i que **RE-MEMBER** hi prova el candidat **5**. A l'esquerra hi representem de nou la situació de partença, a la dreta la solució d'acord amb el patró NATURMÍN que ratifica la validesa d'aquest candidat, i al mig la situació d'emplenament virtual en què **EXPLORA-3*9** fa saltar l'alarma, declara incompatible el candidat **2** a la casella 5,3 i passa a provar el valor **3**, situant l'exploració en el camí correcte, que deambularà sense massa entrebancs fins arribar al final. Si no hagués estat per l'alarma sobre la vulneració de les **normes 2** i **3**, **RE-MEMBER** hauria avançat sense dificultats des de la fila 3ª fins a la 7ª, però un cop allà, hagués hagut d'emprendre una penosíssima retirada, reculant posició a posició fins a situar-se de nou a la casella 5,3.

1 2 0	0 0 0	**5** 0 0	**1 2** 0	0 0 0	**5** 0 0	**1 2** 7	9 6 8	**5** 3 4
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	8 3 4	5 2 1	9 6 7
0 0 **5**	0 **7** 0	0 **1 2**	0 0 **5**	0 **7** 0	0 **1 2**	6 9 **5**	4 **7** 3	8 **1 2**
0 0 0	0 0 **6**	0 0 0	0 0 0	0 0 **6**	0 0 0	9 7 8	3 1 **6**	2 4 5
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	5 6 2	8 4 7	3 9 1
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	3 4 1	2 9 5	6 7 8
0 0 0	0 0 0	0 0 0	7 8 9	1 2 3	0 0 0	7 8 9	1 3 2	4 5 6
0 0 0	0 0 0	0 0 0	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3
0 0 0	**6 5** 0	0 0 0	2 1 3	**6 5** 4	7 8 9	2 1 3	**6 5** 4	7 8 9

Efectivament, així que aquesta casella s'emplena virtualment amb un **3**, **EXPLORA-3*9** detecta sengles infraccions de les **normes 2** i **3** en el requadre 3×3 superior centre (centrat a la casella 5,8), on observant els elements de **R-LLL** (dreta) hi troba 2 llistes que només contenen el valor **8**, infringint la **norma 2**, i un subconjunt de 7 valors (**1**, **2**, **3**, **4**, **5**, **6** i **9**) que només són presents a 6 llistes, infringint la **3**:

(3 8) (4 8) (5 8)	(0 0 **3 4** 0 0 0 **8 9**)	(0 0 **3 4** 0 **6** 0 0 **9**) (0 0 0 0 0 0 0 **8** 0)
(3 7) (4 7) (5 7)	(**1 2 3 4 5** 0 0 **8 9**)	(**1** 0 **3 4** 0 **6** 0 0 **9**) (**1 2** 0 0 **5** 0 0 **8** 0)
(3 6) **7** (5 6)	(0 0 **3 4** 0 0 0 **8 9**)	0 (0 0 0 0 0 0 0 **8** 0)

Com ja suposareu, ens podem permetre el luxe d'especificar que l'emplenament que dispara la primera alarma **EXPLORA-3*9** és a 6,3 (**3**) perquè hem utilitzat una versió trucada de **RE-MEMBER** on savíem en quines condicions es produïa la marrada i l'hem intervinguda perquè **EXPLORA-3*9** sols fos operativa al 100% amb els candidats **2** o **3** a la casella 5,3; altrament, si la plena disponibilitat d'aquesta funció només es donés a l'inici d'una línia (en consonància amb l'últim codi que s'ha presentat), per no llastar massa els temps de cerca, ens hauríem d'haver referit a la casella 1,4 (**3**), tot i que això no afectaria les condicions en què es dispara l'alarma: 4 caselles endavant, els elements de **R-LLL** corresponents al requadre superior centre seguirien sent els mateixos.

Pel que fa a l'opció basada en ampliar **ACTUALITZA-PLUS** a tot l'àmbit de **RE-MEMBER**, més ràpida, la trajectòria la despatxarem en tres representacions, perquè és molt semblant a la del cas precedent, i ens limitarem a la narració escrita per si algú té la paciència de llegir-se-la. La primera aplega el toc final de les dues files inferiors: la casella final de la 1ª fila (**9**) i de la 2ª (**3**), amb la penúltima, 8,2 (**2**), de propina. Aquest inesperat automatisme és degut a la presència del **2** a 9,7: així que haguem emplenat 7,2 (**1**), l'assignació a 9,2 (**3**) queda determinada i **ACTUALITZA-PLUS** actua tapant el forat; després, queda determinada l'assignació a 8,2 (**2**) i la funció torna a actuar. La segona representació aplega el toc final dels dos primers requadres 3×3: l'última casella del primer (**9**) i del segon (**3**), amb la casella 6,7 (**8**) de propina. Aquest inesperat automatisme ara és degut a la presència del **7** a 5,7: així que haguem emplenat 5,3 (**2**), l'assignació a 6,3 (**3**) queda determinada i **ACTUALITZA-PLUS** actua tapant el forat. Després d'això, ¿hem de repetir que l'assignació a 6,7 (**8**) queda determinada i la funció torna a actuar?: doncs, no exactament. Si penseu en quins són els elements de **R-LLL** corresponents a les caselles 6,7 i 6,9, just després que **ACTUALITZA-PLUS** hagi emplenat 6,3 (**4**), us trobareu dues llistes **'(nil nil nil nil nil nil nil 8 nil)**. **ACTUALITZA-PLUS** podria emplenar-ne una qualsevol, però com que la cerca es fa ordenadament per files, li tocarà a la primera: l'element de **R-LLL** associat passarà a **nil** i el corresponent a la posició 6,9 es transformarà en **'(nil nil nil nil nil nil nil nil nil)** provocant la fi de la progressió, l'anul·lació de l'ocupació 6,3 (**3**) i les seves seqüeles i, com que ja no queden candidats, el retorn a la casella precedent per provar amb **3**.

1 2 0 0 0 0 0 0 **5**	0 0 0 0 0 0 0 **7** 0	**5** 0 0 0 0 0 0 **1 2**	**1 2** 0 0 0 0 0 0 **5**	0 0 X 0 0 0 0 **7** 8
5 0 0 0 0 0 0 **1 2**	**1 2** 7 8 3 4 6 9 **5**	9 6 8 5 2 1 4 **7** 3	**5** 3 4 9 6 7 8 **1 2**	
0 0 0 0 0 0 0 0 0	0 0 **6** 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 **6** 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	9 7 8 5 6 2 3 4 1	3 1 **6** 8 4 7 2 9 5	2 4 5 3 9 1 6 7 8	
0 0 0 4 5 6 2 1 3	0 0 0 7 8 9 **6 5** 4	0 0 0 1 2 3 7 8 9	7 8 9 4 5 6 2 1 3	0 0 0 1 2 3 7 8 9
7 8 9 4 5 6 2 1 3	1 3 2 7 8 9 **6 5** 4	4 5 6 1 2 3 7 8 9		

La tercera representació resumeix la resta de l'exploració des d'aquest moment, que amb l'ajut d'**ACTUALITZA-PLUS** es converteix en una marxa triomfal: el toc final del segon i tercer requadre 3×3, amb l'emplenament automàtic de 6,3 (**2**) i 9,3 (**6**); l'emplenament de 4,4 (**2**) desencadena l'emplenament automàtic de 5,4 (**9**), 5,8 (**2**), 6,8 (**1**), 4,8 (**5**) i 5,9 (**6**); l'emplenament de 6,4 (**5**) desencadena el de 6,5 (**7**); l'emplenament de 7,4 (**6**) desferma el de 8,4 (**7**), 9,4 (**8**) i 8,8 (**6**); l'emplenament de 1,5 (**5**) desencadena la seqüència d'emplenaments automàtics 2,5 (**6**), 9,6 (**5**), 9,5 (**1**), 5,5 (**4**), 5,6 (**1**), 8,6 (**4**) i 1,7 (**6**); l'emplenament de 3,5 (**2**) desencadena el de 7,6 (**2**); l'emplenament de 4,5 (**8**) desferma el de 4,6 (**3**); l'emplenament de 7,5 (**3**) desencadena la seqüència final d'emplenaments automàtics 8,5 (**9**), 8,9 (**3**), 6,9 (**8**), 6,7 (**3**), 2,7 (**9**), 2,6 (**7**), 3,6 (**8**), 1,6 (**9**), 4,7 (**4**), 7,7 (**8**), 1,8 (**8**), 2,8 (**3**), 7,8 (**9**), 4,9 (**9**), 3,8 (**4**), 9,8 (**7**), 3,9 (**7**) i 9,9 (**4**).

Però en el tercer i últim dels escenaris escollits, continuació del segon en què la casella 5,1 (**5**) ja s'ha consolidat, hem proseguit emplenant 8,2 (**8**), 6,4 (**2**) i 8,6 (**7**), i **RE-MEMBER** es disposa a provar el candidat **5** a la casella 2,6, veurem que la versió equipada amb **EXPLORA-3*9** hi té ben poca cosa a fer, i només la que incorpora **ACTUALITZA-PLUS** aconsegueix que el veredicte (favorable) per a aquest candidat arribi en un temps raonable. Justificarem el perquè de cada resultat i, començant per la primera versió, presentem una sèrie de 9×3 = 27 representacions:

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
0 0 0	0 0 0	0 0 0
3 4 1	5 0 **2**	0 0 0
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
6 7 2	4 0 0	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
6 7 2	5 3 1	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
6 7 2	9 0 0	0 0 0
3 4 1	5 0 **2**	0 0 0
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
6 7 8	4 0 0	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
6 7 8	5 3 1	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
6 9 0	0 0 0	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
7 9 2	0 0 0	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
7 9 6	4 0 0	0 0 0
3 4 1	5 0 **2**	0 0 0
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
7 9 6	5 3 1	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
8 7 6	4 0 0	0 0 0
3 4 1	5 0 **2**	0 0 0
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
8 7 6	5 3 1	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

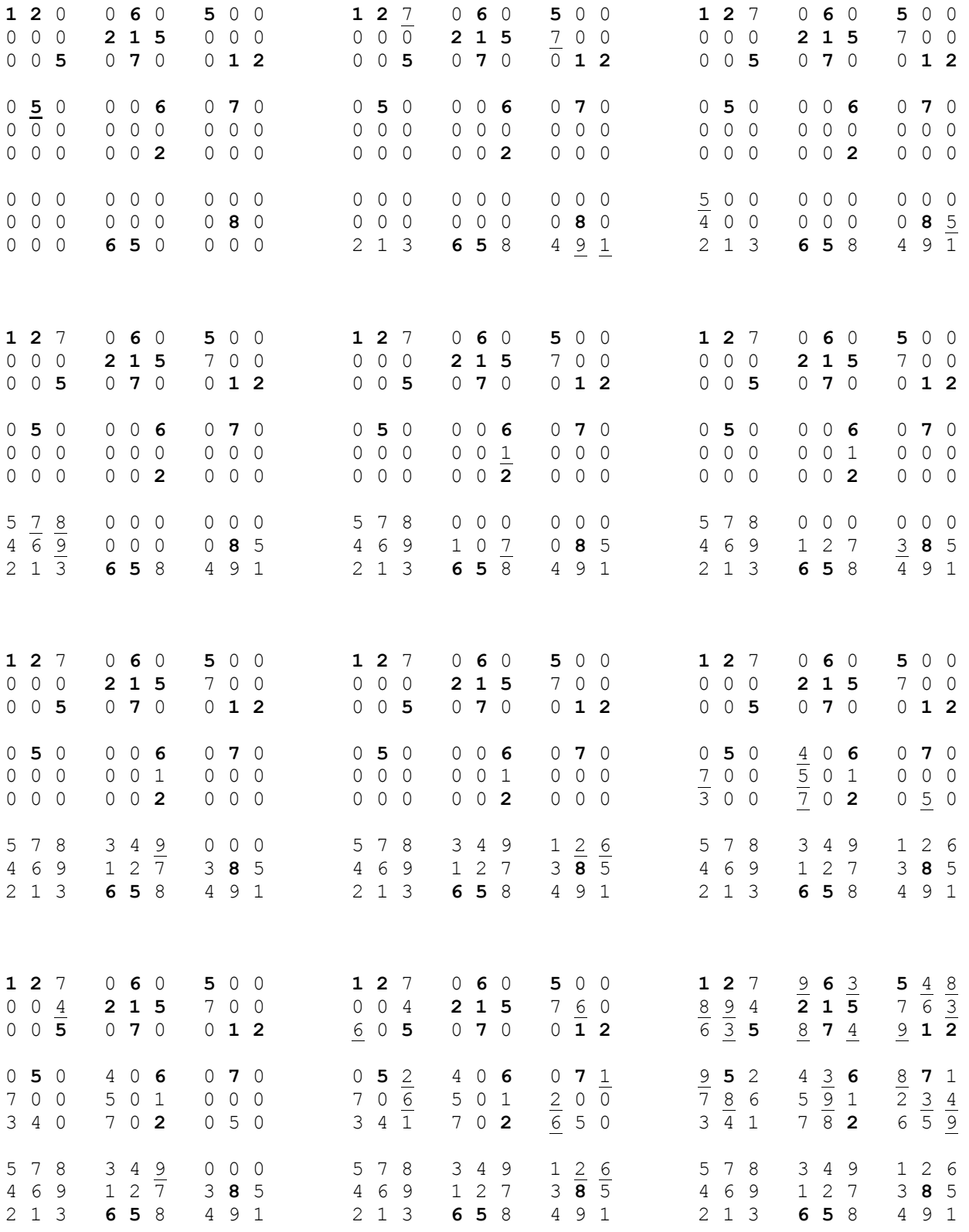
1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
8 7 6	9 0 0	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 2 0	0 **6** 0	**5** 0 0
0 0 0	**2 1 5**	0 0 0
0 0 **5**	0 **7** 0	0 **1 2**
0 **5** 0	0 0 **6**	0 **7** 0
9 7 2	0 0 0	0 0 0
3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5
2 1 3	**6 5** 8	4 9 7

1 **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0
0 0 0	**2** **1** **5**	0 0 0	0 0 0	**2** **1** **5**	0 0 0	0 0 0	**2** **1** **5**	0 0 0
0 0 **5**	0 **7** 0	0 **1** **2**	0 0 **5**	0 **7** 0	0 **1** **2**	0 0 **5**	0 **7** 0	0 **1** **2**
0 **5** 0	0 0 **6**	0 **7** 0	0 **5** 0	0 0 **6**	0 **7** 0	0 **5** 0	0 0 **6**	0 **7** 0
9 7 6	4 0 0	0 0 0	9 7 6	5 3 1	0 0 0	6 7 2	4 0 0	0 0 0
3 4 1	7 8 **2**	0 0 0	3 4 1	7 8 **2**	6 5 9	3 4 1	7 9 **2**	6 5 8
5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5
2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7
1 **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0
0 0 0	**2** **1** **5**	0 0 0	0 0 0	**2** **1** **5**	0 0 0	0 0 0	**2** **1** **5**	0 0 0
0 0 **5**	0 **7** 0	0 **1** **2**	0 0 **5**	0 **7** 0	0 **1** **2**	0 0 **5**	0 **7** 0	0 **1** **2**
0 **5** 0	0 0 **6**	0 **7** 0	0 **5** 0	0 0 **6**	0 **7** 0	0 **5** 0	0 0 **6**	0 **7** 0
6 7 2	5 3 1	0 0 0	6 7 2	8 0 0	0 0 0	6 7 8	4 0 0	0 0 0
3 4 1	7 8 **2**	0 0 0	3 4 1	7 8 **2**	6 5 9	3 4 1	7 9 **2**	6 5 8
5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5
2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7
1 **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0
0 0 0	**2** **1** **5**	0 0 0	0 0 0	**2** **1** **5**	0 0 0	0 0 0	**2** **1** **5**	0 0 0
0 0 **5**	0 **7** 0	0 **1** **2**	0 0 **5**	0 **7** 0	0 **1** **2**	0 0 **5**	0 **7** 0	0 **1** **2**
0 **5** 0	0 0 **6**	0 **7** 0	0 **5** 0	0 0 **6**	0 **7** 0	0 **5** 0	0 0 **6**	0 **7** 0
6 7 8	5 3 1	0 0 0	6 9 0	0 0 0	0 0 0	7 9 6	4 0 0	0 0 0
3 4 1	7 8 **2**	0 0 0	3 4 1	7 8 **2**	6 5 9	3 4 1	7 9 **2**	6 5 8
5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6
4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5
2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7
1 **2** 0	0 **6** 0	**5** 0 0	**1** **2** 0	0 **6** 0	**5** 0 0	**1** **2** 7	9 **6** 3	**5** 4 8
0 0 0	**2** **1** **5**	0 0 0	9 7 4	**2** **1** **5**	X 0 0	8 9 4	**2** **1** **5**	7 6 3
0 0 **5**	0 **7** 0	0 **1** **2**	6 3 **5**	9 **7** 4	8 **1** **2**	6 3 **5**	8 **7** 4	9 **1** **2**
0 **5** 0	0 0 **6**	0 **7** 0	8 **5** 2	4 3 **6**	9 **7** 1	9 **5** 2	4 3 **6**	8 **7** 1
7 9 6	5 3 1	0 0 0	7 9 6	5 8 1	2 3 4	7 8 6	5 9 1	2 3 4
3 4 1	7 8 **2**	0 0 0	3 4 1	7 8 **2**	6 5 9	3 4 1	7 8 **2**	6 5 9
5 8 9	3 4 7	1 2 6	5 8 9	3 4 7	1 2 6	5 7 8	3 4 9	1 2 6
4 6 7	1 2 9	3 **8** 5	4 6 7	1 2 9	3 **8** 5	4 6 9	1 2 7	3 **8** 5
2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7	2 1 3	**6** **5** 8	4 9 7

En aquesta llarga seqüència, les primeres 25 representacions reflecteixen l'estat d'emplenament virtual de les situacions en què **EXPLORA-3*9**, integrada a **RE-MEMBER** en règim intensiu, detecta alguna infracció de les **normes 1** o **2**/**3** (a l'esquerra de l'última renglera hi tenim la 25ª) abans d'arribar a l'estat d'emplenament en fals més avançat (última renglera, centre). L'aclariment "en fals" vol dir que no estem en el camí correcte, perquè l'emplenament virtual de l'escaquer que certificarà la viabilitat del candidat **5** a la casella 2,6 (última renglera, dreta) té un **9** a 3,2 i un **7** a 6,2, mentre que en les situacions precedents representades aquests valors estan permutats. Dit d'una altra forma: un **7** a 3,2 no vulnera cap norma i per això li passa per alt a **EXPLORA-3*9**. És a partir de la 4ª fila que la funció hi intervé accelerant una progressió en fals que, en quedar blocada a 7,8 (X, en el centre), iniciarà la retirada fins a 3,2, feixuga malgrat la participació d' **EXPLORA-3*9**.

Veiem com les coses li van millor a **RE-MEMBER** si pot comptar amb **ACTUALITZA-PLUS.**



Com que la seqüència que descriu l'emplenament virtual de l'escaquer (a càrrec de la versió de **RE-MEMBER** equipada amb **ACTUALITZA-PLUS**) també és llarga, s'ha estimat oportú obrir-la amb la representació de la situació inicial, amb els emplenaments previs en negreta, igual que el valor **5** de prova en la casella 2,6, que a més hem subratllat (1ª renglera, esquerra); en canvi no s'han subratllat les caselles 6,8 (**5**) i 5,9 (**6**) que, si heu seguit l'emplenament en els dos casos precedents, sabreu que s'emplenen automàticament. Diem això perquè les 11 representacions restants sí que mantenen el conveni utilitzat en aquests dos casos: subratllar cada cadena de forats tapats per **ACTUALITZA-PLUS** a conseqüència de l'últim emplenament manual. Començant per la segona (1ª renglera, centre), el previsible emplenament automàtic de les dues caselles finals de la 1ª fila (**7** i **9**) no s'atura aquí sinó que segueix amb el de 7,8 (**7**) i 3,9 (**7**): atenció, perquè aquest últim emplenament serà el que

condicionarà el de la casella 3,2 (**9**), en franca divergència amb l'assignació que la versió alternativa de **RE-MEMBER** havia realitzat (**7**) obeint cegament la dinàmica NATURMÍN. Però no avancem esdeveniments i passem al següent emplenament ordinari, 1,2 (**4**), que arrossegarà els automàtics de 1,3 (**5**) i 9,2 (**5**) (1ª renglera, dreta), i el de 2,2 (**6**) que, amb l'emplenament automàtic de 3,2 (**9**), 3,3 (**8**) i 3,2 (**7**), completarà el del primer requadre 3×3 (2ª renglera, esquerra).

Fixeu-vos d'on surt la primera divergència amb la trajectòria d'exploració traçada per la versió alternativa de **RE-MEMBER** (el **7** a 3,2 i el **9** a 6,2, valors que ara es permuten): del **7** situat a 9,1, que determina l'aparició del mateix valor a 7,8 i, mitjançant aquest, a 3,9; la primera posició constituïa la barrera infranquejable per a aquella **RE-MEMBER**, infranquejable almenys sense abans haver retrocedit a la casella 3,2 i haver canviat el **7** pel **9** (per imperatiu d'un **7** que ja tenia un lloc reservat en la segona posició, malgrat que ella ho desconegués). El problema era que cap dispositiu (ni, en particular, l'**EXPLORA-3*9** plenipotenciari) no va vetar el **7** a 3,2 ni cap altra ocupació que hagués frenat l'avanç gairebé imparable fins al galdós bloqueig de 7,8, propiciant una retirada abans. Per contra, la versió bona de **RE-MEMBER** haurà aconseguit completar l'emplenament de l'escaquer en només 15 jugades (naturalment, parlem dels emplenaments fa poc qualificats d'ordinaris, contraposant-los als automàtics). Com que ja en portem 7, veiem les que falten.

Seguirem amb l'emplenament de la casella 4,2 (**1**), que provoca el de 6,2 (**7**) i 6,5 (**1**) (2ª renglera, centre); el de la casella 5,2 (**2**), que provoca el de 6,2 (**3**) (2ª renglera, dreta); el de 5,3 (**4**), que provoca el de 6,3 (**9**) (3ª renglera esquerra); el de 7,3 (**1**), que provoca el de 9,3 (**6**) i 8,3 (**2**) (3ª renglera, centre); el de 1,4 (**3**), que desencadena una seqüència d'emplenaments automàtics, 1,5 (**7**), 4,4 (**7**), 4,5 (**5**), 4,6 (**4**) i 8,4 (**5**) (3ª renglera, dreta); el de 2,4 (**4**), que provoca el de 3,8 (**4**) (4ª renglera, esquerra); el de 3,4 (**1**), que desferma una seqüència d'emplenaments automàtics, 3,6 (**2**), 3,5 (**6**), 9,6 (**1**), 7,5 (**2**), 7,4 (**6**), 8,8 (**6**) i 1,7 (**6**) (4ª renglera, centre). Finalment, l'últim emplenament ordinari és el de la casella 5,4 (**8**), que desferma l'emplenament automàtic de la resta, 9,4 (**9**), 7,6 (**8**), 1,6 (**9**), 2,5 (**8**), 5,6 (**3**), 5,5 (**9**), 7,7 (**9**), 2,7 (**3**), 4,7 (**8**), 6,7 (**4**), 1,8 (**8**), 2,8 (**9**), 9,8 (**3**), 9,5 (**4**), 8,5 (**3**), 4,9 (**9**), 6,9 (**3**), 8,9 (**4**) i 9,9 (**8**) (4ª renglera, dreta). Com abans, en un tres i no res aquest **RE-MEMBER** ha deixat ple l'escaquer, mentre que la versió precedent amb **EXPLORA-3*9** no ha aconseguit fer caure el temps d'espera fins un nivell acceptable, i no ens referim únicament a la versió *heavy* (**EXPLORA-3*9** intervenint a totes les caselles) que hem utilitzat abans per poder-vos oferir les 9×3 = 27 representacions (això sí, trucant-la per restringir-la a només una etapa de l'exploració i fer la experiència suportable), sinó a la versió *light* (penúltim codi: intervenció limitada als canvis de fila).

Amb la tranquil·litat d'haver minimitzat el risc de pauses anormalment llargues (fixeu-vos que parlem de minimitzar i no d'haver-lo eliminat del tot), repetirem dues de les conclusions que ja havíem avançat en obrir la pàgina 197: **EXPLORA-3*9** (reduïda definitivament a supervisar la **norma 1**, tret de la recursió **INI**cial) es limitarà als requadres 3×3; tan per escurçar les esperes que, de mitjana, encara pequen de llargues, com per esmorteir l'impacte de les que puguin anar més enllà, malgrat les cauteles adoptades, mantindrem el dispositiu previst perquè durant tot l'emplenament el sudoku estigui ordenat de més a menys ocupació. La primera mesura es justifica perquè en cap dels assajos del present capítol ens hem trobat amb la necessitat d'ampliar les exploracions a files i columnes (potser la memòria d'un cas del capítol precedent en què també calia explorar files i columnes per trobar la infracció, no documentada, no té consistència i només es tractava d'un error), però si no hagués estat així igualment caldria fer de la necessitat virtut, perquè no ens podem permetre multiplicar quasi per 3 la cadència d'execucions **RE-MEMBER.** Però adoptar la segona, així sense més, ens agafa ja massa escarmentats de fer el ridícul veient com s'ensorraven castells de cartes muntats sobre pressupostos que havíem acceptat acríticament com "evidents": ¿qui ens diu, per exemple, que encara es mantindrà la relació de retrocessos enregistrats abans de les últimes esmenes?; ¿que no podria ser que s'hagués invertit, posats en la hipòtesi més desfavorable?. Justament per tancar aquest subcapítol amb les màximes garanties, repetirem les 3 sèries de recomptes amb què ens havíem embrancat (3×3 = 9 taules), no sempre sense poder-los completar, a causa de les enganxades que ara ja creiem tenir superades. A més, com que ja no canviarem res més a **RE-MEMBER**, ho farem amb NATURMÍN i també amb NATURMAX. Girant full teniu representats els requadres 9×3 de la primera, i tres pàgines més endavant, els de la segona solució. Igual que abans, les taules es refereixen a emplenaments en cadascun dels requadres: individual o conjunt de les 10 caselles destacades en negreta, o emplenament del requadre sencer.

```
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       9 7 8   5 3 1   6 4 2
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       6 4 2   9 7 8   5 3 1
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       5 3 1   6 4 2   9 7 8

0 0 0   0 0 0   0 0 0       8 9 7   2 1 4   3 6 5       0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0       3 6 5   8 9 7   2 1 4       0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0       2 1 4   3 6 5   8 9 7       0 0 0   0 0 0   0 0 0

7 8 9   1 2 3   4 5 6       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
4 5 6   7 8 9   1 2 3       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
1 2 3   4 5 6   7 8 9       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
```

Exploració caselles 1r requadre amb l'escaquer buit, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà	**1.023**	+ **59**	+ **9**	= **1.091**	retrocessos
- Si només omplim 5,1 (**5**) hi haurà	**822**	+ **37**	+ **2**	= **861**	retrocessos
- Si només omplim 9,1 (**9**) hi haurà	**510**	+ **65**	+ **0**	= **575**	retrocessos
- Si només omplim 2,2 (**5**) hi haurà	**822**	+ **90**	+ **6**	= **918**	retrocessos
- Si només omplim 4,2 (**7**) hi haurà	**372**	+ **60**	+ **0**	= **432**	retrocessos
- Si només omplim 6,2 (**9**) hi haurà	**372**	+ **39**	+ **0**	= **411**	retrocessos
- Si només omplim 8,2 (**2**) hi haurà	**60**	+ **113**	+ **0**	= **173**	retrocessos
- Si només omplim 1,3 (**7**) hi haurà	**357**	+ **119**	+ **0**	= **476**	retrocessos
- Si només omplim 5,3 (**2**) hi haurà	**354**	+ **81**	+ **1**	= **436**	retrocessos
- Si només omplim 9,3 (**6**) hi haurà	**1.149**	+ **92**	+ **0**	= **1.241**	retrocessos
Sumant retrocessos per requadres:	**5.841**	+ **755**	+ **18**	= **6.614**	retrocessos

Exploració caselles 2n requadre amb l'escaquer buit, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà	**1.134**	+ **46**	+ **28**	= **1.208**	retrocessos
- Si només omplim 5,4 (**6**) hi haurà	**1.020**	+ **34**	+ **0**	= **1.054**	retrocessos
- Si només omplim 9,4 (**7**) hi haurà	**900**	+ **31**	+ **1**	= **932**	retrocessos
- Si només omplim 2,5 (**6**) hi haurà	**1.134**	+ **49**	+ **2**	= **1.185**	retrocessos
- Si només omplim 4,5 (**8**) hi haurà	**1.044**	+ **39**	+ **0**	= **1.083**	retrocessos
- Si només omplim 6,5 (**7**) hi haurà	**935**	+ **16**	+ **1**	= **952**	retrocessos
- Si només omplim 8,5 (**1**) hi haurà	**900**	+ **14**	+ **0**	= **914**	retrocessos
- Si només omplim 1,6 (**8**) hi haurà	**1.134**	+ **0**	+ **1**	= **1.135**	retrocessos
- Si només omplim 5,6 (**1**) hi haurà	**1.020**	+ **20**	+ **0**	= **1.040**	retrocessos
- Si només omplim 9,6 (**5**) hi haurà	**900**	+ **54**	+ **1**	= **955**	retrocessos
Sumant retrocessos per requadres:	**10.121**	+ **303**	+ **34**	= **10.458**	retrocessos

Exploració caselles 3r requadre amb l'escaquer buit, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**1.134**	+ **36**	+ **14**	= **1.184**	retrocessos
- Si només omplim 5,7 (**4**) hi haurà	**1.020**	+ **41**	+ **3**	= **1.064**	retrocessos
- Si només omplim 9,7 (**8**) hi haurà	**900**	+ **39**	+ **1**	= **940**	retrocessos
- Si només omplim 2,8 (**4**) hi haurà	**1.134**	+ **22**	+ **2**	= **1.158**	retrocessos
- Si només omplim 4,8 (**9**) hi haurà	**1.044**	+ **41**	+ **0**	= **1.085**	retrocessos
- Si només omplim 6,8 (**8**) hi haurà	**935**	+ **51**	+ **1**	= **987**	retrocessos
- Si només omplim 8,8 (**3**) hi haurà	**900**	+ **49**	+ **0**	= **949**	retrocessos
- Si només omplim 1,9 (**9**) hi haurà	**1.134**	+ **36**	+ **5**	= **1.175**	retrocessos
- Si només omplim 5,9 (**3**) hi haurà	**1.020**	+ **41**	+ **0**	= **1.061**	retrocessos
- Si només omplim 9,9 (**2**) hi haurà	**900**	+ **39**	+ **1**	= **940**	retrocessos
Sumant retrocessos per requadres:	**10.121**	+ **395**	+ **27**	= **10.543**	retrocessos

Exploració i emplenament de 10 caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà	**1.023**	+ **59**	+ **9**	= **1.091**	retrocessos
- Si també omplim 5,1 (**5**) hi haurà	**696**	+ **35**	+ **0**	= **731**	retrocessos
- Si també omplim 9,1 (**9**) hi haurà	**466**	+ **47**	+ **1**	= **514**	retrocessos
- Si també omplim 2,2 (**5**) hi haurà	**696**	+ **83**	+ **4**	= **783**	retrocessos
- Si també omplim 4,2 (**7**) hi haurà	**336**	+ **58**	+ **0**	= **394**	retrocessos
- Si també omplim 6,2 (**9**) hi haurà	**81**	+ **31**	+ **0**	= **112**	retrocessos
- Si també omplim 8,2 (**2**) hi haurà	**7**	+ **36**	+ **0**	= **43**	retrocessos

- Si també omplim 1,3 (**7**) hi haurà	**4**	+	**43**	+	**0**	= **47** retrocessos
- Si també omplim 5,3 (**2**) hi haurà	**4**	+	**39**	+	**0**	= **43** retrocessos
- Si també omplim 9,3 (**6**) hi haurà	**11**	+	**76**	+	**0**	= **87** retrocessos
Sumant retrocessos per requadres:	**3.324**	+	**507**	+	**14**	= **3.845** retrocessos

Exploració i emplenament de 10 caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà	**1.134**	+	**46**	+	**28**	= **1.208** retrocessos
- Si també omplim 5,4 (**6**) hi haurà	**905**	+	**32**	+	**0**	= **937** retrocessos
- Si també omplim 9,4 (**7**) hi haurà	**726**	+	**27**	+	**1**	= **754** retrocessos
- Si també omplim 2,5 (**6**) hi haurà	**1.008**	+	**41**	+	**1**	= **1.050** retrocessos
- Si també omplim 4,5 (**8**) hi haurà	**918**	+	**35**	+	**2**	= **955** retrocessos
- Si també omplim 6,5 (**7**) hi haurà	**814**	+	**10**	+	**1**	= **825** retrocessos
- Si també omplim 8,5 (**1**) hi haurà	**450**	+	**0**	+	**0**	= **450** retrocessos
- Si també omplim 1,6 (**8**) hi haurà	**294**	+	**0**	+	**0**	= **294** retrocessos
- Si també omplim 5,6 (**1**) hi haurà	**195**	+	**1**	+	**1**	= **197** retrocessos
- Si també omplim 9,6 (**5**) hi haurà	**147**	+	**0**	+	**0**	= **147** retrocessos
Sumant retrocessos per requadres:	**6.591**	+	**192**	+	**34**	= **6.817** retrocessos

Exploració i emplenament de 10 caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**1.134**	+	**36**	+	**14**	= **1.184** retrocessos
- Si també omplim 5,7 (**4**) hi haurà	**894**	+	**39**	+	**3**	= **936** retrocessos
- Si també omplim 9,7 (**8**) hi haurà	**648**	+	**31**	+	**0**	= **679** retrocessos
- Si també omplim 2,8 (**4**) hi haurà	**1.008**	+	**30**	+	**0**	= **1.038** retrocessos
- Si també omplim 4,8 (**9**) hi haurà	**918**	+	**32**	+	**0**	= **950** retrocessos
- Si també omplim 6,8 (**8**) hi haurà	**700**	+	**17**	+	**0**	= **717** retrocessos
- Si també omplim 8,8 (**3**) hi haurà	**414**	+	**16**	+	**0**	= **430** retrocessos
- Si també omplim 1,9 (**9**) hi haurà	**273**	+	**10**	+	**0**	= **283** retrocessos
- Si també omplim 5,9 (**3**) hi haurà	**218**	+	**5**	+	**0**	= **223** retrocessos
- Si també omplim 9,9 (**2**) hi haurà	**148**	+	**0**	+	**0**	= **148** retrocessos
Sumant retrocessos per requadres:	**6.355**	+	**216**	+	**17**	= **6.588** retrocessos

Exploració i emplenament de les caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**1**) hi haurà	**1.023**	+	**59**	+	**9**	= **1.091** retrocessos
- Si també omplim 2,1 (**2**) hi haurà	**897**	+	**36**	+	**11**	= **944** retrocessos
- Si també omplim 3,1 (**3**) hi haurà	**771**	+	**42**	+	**0**	= **813** retrocessos
- Si també omplim 4,1 (**4**) hi haurà	**444**	+	**31**	+	**2**	= **477** retrocessos
- Si també omplim 5,1 (**5**) hi haurà	**318**	+	**29**	+	**0**	= **347** retrocessos
- Si també omplim 6,1 (**6**) hi haurà	**192**	+	**49**	+	**0**	= **241** retrocessos
- Si també omplim 7,1 (**7**) hi haurà	**378**	+	**21**	+	**0**	= **399** retrocessos
- Si també omplim 8,1 (**8**) hi haurà	**252**	+	**14**	+	**0**	= **266** retrocessos
i també s'omple 9,1 (**9**)						
- Si també omplim 1,2 (**4**) hi haurà	**444**	+	**81**	+	**5**	= **530** retrocessos
- Si també omplim 2,2 (**5**) hi haurà	**318**	+	**62**	+	**0**	= **380** retrocessos
- Si també omplim 3,2 (**6**) hi haurà	**192**	+	**58**	+	**0**	= **250** retrocessos
- Si també omplim 4,2 (**7**) hi haurà	**108**	+	**18**	+	**0**	= **126** retrocessos
- Si també omplim 5,2 (**8**) hi haurà	**18**	+	**11**	+	**0**	= **29** retrocessos
- Si també omplim 6,2 (**9**) hi haurà	**0**	+	**7**	+	**0**	= **7** retrocessos
- Si també omplim 7,2 (**1**) hi haurà	**0**	+	**22**	+	**0**	= **22** retrocessos
- Si també omplim 8,2 (**2**) hi haurà	**0**	+	**14**	+	**0**	= **14** retrocessos
i també s'omple 9,2 (**3**)						
- Si també omplim 1,3 (**7**) hi haurà	**0**	+	**21**	+	**0**	= **21** retrocessos
- Si també omplim 2,3 (**8**) hi haurà	**0**	+	**14**	+	**0**	= **14** retrocessos
i també s'omple 3,3 (**9**)						
- Si també omplim 4,3 (**1**) hi haurà	**0**	+	**15**	+	**0**	= **15** retrocessos
- Si també omplim 5,3 (**2**) hi haurà	**0**	+	**14**	+	**0**	= **14** retrocessos
i també s'omple 6,3 (**3**)						
- Si també omplim 7,3 (**4**) hi haurà	**0**	+	**19**	+	**0**	= **19** retrocessos
- Si també omplim 8,3 (**5**) hi haurà	**0**	+	**14**	+	**0**	= **14** retrocessos
i també s'omple 9,3 (**6**)						
Sumant retrocessos per requadres:	**5.355**	+	**651**	+	**27**	= **6.033** retrocessos

Exploració i emplenament de les caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**2**) hi haurà	**1.134**	+	**46**	+	**28**	= **1.208** retrocessos
- Si també omplim 2,4 (**1**) hi haurà	**1.008**	+	**40**	+	**5**	= **1.053** retrocessos
- Si també omplim 3,4 (**4**) hi haurà	**778**	+	**28**	+	**2**	= **808** retrocessos
- Si també omplim 4,4 (**3**) hi haurà	**726**	+	**28**	+	**2**	= **756** retrocessos
- Si també omplim 5,4 (**6**) hi haurà	**426**	+	**24**	+	**0**	= **450** retrocessos
- Si també omplim 6,4 (**5**) hi haurà	**340**	+	**23**	+	**1**	= **364** retrocessos
- Si també omplim 7,4 (**8**) hi haurà	**288**	+	**21**	+	**0**	= **309** retrocessos
- Si també omplim 8,4 (**9**) hi haurà	**192**	+	**14**	+	**0**	= **206** retrocessos
i també s'omple 9,4 (**7**)						
- Si també omplim 1,5 (**3**) hi haurà	**576**	+	**37**	+	**3**	= **616** retrocessos
- Si també omplim 2,5 (**6**) hi haurà	**480**	+	**32**	+	**0**	= **512** retrocessos
- Si també omplim 3,5 (**5**) hi haurà	**384**	+	**19**	+	**1**	= **404** retrocessos
- Si també omplim 4,5 (**8**) hi haurà	**288**	+	**18**	+	**0**	= **306** retrocessos
- Si també omplim 5,5 (**9**) hi haurà	**1.686**	+	**1.588**	+	**0**	= **3.274** retrocessos
- Si també omplim 6,5 (**7**) hi haurà	**52**	+	**0**	+	**0**	= **52** retrocessos
- Si també omplim 7,5 (**2**) hi haurà	**102**	+	**0**	+	**0**	= **102** retrocessos
- Si també omplim 8,5 (**1**) hi haurà	**16**	+	**0**	+	**0**	= **16** retrocessos
i també s'omple 9,5 (**4**)						
- Si també omplim 1,6 (**8**) hi haurà	**24**	+	**0**	+	**0**	= **24** retrocessos
- Si també omplim 2,6 (**9**) hi haurà	**16**	+	**0**	+	**0**	= **16** retrocessos
i també s'omple 3,6 (**7**)						
- Si també omplim 4,6 (**2**) hi haurà	**22**	+	**0**	+	**0**	= **22** retrocessos
- Si també omplim 5,6 (**1**) hi haurà	**12**	+	**0**	+	**0**	= **12** retrocessos
- i també s'omple 6,6 (**4**)						
- Si també omplim 7,6 (**3**) hi haurà	**16**	+	**0**	+	**0**	= **16** retrocessos
- Si també omplim 8,6 (**6**) hi haurà	**8**	+	**0**	+	**0**	= **8** retrocessos
i també s'omple 9,6 (**5**)						
Sumant retrocessos per requadres:	**8.574**	+	**1.918**	+	**42**	= **10.534** retrocessos

Exploració i emplenament de les caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**1.134**	+	**36**	+	**14**	= **1.184** retrocessos
- Si també omplim 2,7 (**3**) hi haurà	**1.008**	+	**18**	+	**4**	= **1.030** retrocessos
- Si també omplim 3,7 (**1**) hi haurà	**778**	+	**30**	+	**0**	= **808** retrocessos
- Si també omplim 4,7 (**6**) hi haurà	**726**	+	**25**	+	**0**	= **751** retrocessos
- Si també omplim 5,7 (**4**) hi haurà	**538**	+	**15**	+	**0**	= **553** retrocessos
- Si també omplim 6,7 (**2**) hi haurà	**431**	+	**18**	+	**0**	= **449** retrocessos
- Si també omplim 7,7 (**9**) hi haurà	**345**	+	**12**	+	**0**	= **357** retrocessos
- Si també omplim 8,7 (**7**) hi haurà	**230**	+	**8**	+	**0**	= **238** retrocessos
i també s'omple 9,7 (**8**)						
- Si també omplim 1,8 (**6**) hi haurà	**690**	+	**14**	+	**0**	= **704** retrocessos
- Si també omplim 2,8 (**4**) hi haurà	**565**	+	**9**	+	**0**	= **574** retrocessos
- Si també omplim 3,8 (**2**) hi haurà	**460**	+	**0**	+	**0**	= **460** retrocessos
- Si també omplim 4,8 (**9**) hi haurà	**345**	+	**0**	+	**0**	= **345** retrocessos
- Si també omplim 5,8 (**7**) hi haurà	**180**	+	**0**	+	**0**	= **180** retrocessos
- Si també omplim 6,8 (**8**) hi haurà	**72**	+	**0**	+	**0**	= **72** retrocessos
- Si també omplim 7,8 (**5**) hi haurà	**130**	+	**0**	+	**0**	= **130** retrocessos
- Si també omplim 8,8 (**3**) hi haurà	**16**	+	**0**	+	**0**	= **16** retrocessos
i també s'omple 9,8 (**1**)						
- Si també omplim 1,9 (**9**) hi haurà	**24**	+	**0**	+	**0**	= **24** retrocessos
- Si també omplim 2,9 (**7**) hi haurà	**16**	+	**0**	+	**0**	= **16** retrocessos
i també s'omple 3,9 (**8**)						
- Si també omplim 4,9 (**5**) hi haurà	**20**	+	**0**	+	**0**	= **20** retrocessos
- Si també omplim 5,9 (**3**) hi haurà	**12**	+	**0**	+	**0**	= **12** retrocessos
- i també s'omple 6,9 (**1**)						
- Si també omplim 7,9 (**6**) hi haurà	**18**	+	**0**	+	**0**	= **18** retrocessos
- Si també omplim 8,9 (**4**) hi haurà	**8**	+	**0**	+	**0**	= **8** retrocessos
i també s'omple 9,9 (**2**)						
Sumant retrocessos per requadres:	**7.746**	+	**185**	+	**18**	= **7.949** retrocessos

Com hem vist abans, la major incidència dels retrocessos (que sempre repercuteixen més en el primer requadre 9×3) es dóna clarament en el segon i tercer requadre, la qual cosa vol dir que convé realitzar una permutació per deixar el més poblat a la

part inferior de l'escaquer. Però, ¿què fem amb el segon més poblat?: ¿el passem al mig o a la part superior?. En aquest extrem la situació no s'acaba de definir, perquè hi ha una mena d'empat tècnic entre els requadres 9×3 central i superior, fins i tot en l'últim cas (emplenament total de cada requadre): si de la penúltima taula en traguéssim l'excés que correspon a l'emplenament de la casella 5,5 (**9**), **3.274** (més endavant discutirem si fer això era pertinent), deixant-ho en **274**, per exemple, el total de retrocessos del 2n requadre quedaria en **7.534**, molt semblant als **7.949** retrocessos del 3r. Només per forçar el desempat ja valia la pena passar a NATURMAX i repetir experiències i taules, tret de les tres primeres, que serien idèntiques a les de NATURMÍN: els retrocessos es produeixen en activar la casella, responen al conjunt de candidats avaluats i l'escollit només condiciona l'avenir.

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
1 3 2 4 **6** 8 **5** 7 9	5 **7** 9 **1** 3 **2** 4 **6** 8	4 6 **8** 5 **7** 9 1 3 **2**			
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	**2** 1 3 7 **4** 5 **8** 9 6	8 **9** 6 **2** 1 **3** 7 **4** 5	7 4 **5** 8 **9** 6 2 1 **3**
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0			
3 2 1 6 **5** 4 **9** 8 7	9 **8** 7 **3** 2 **1** 6 **5** 4	6 5 **4** 9 **8** 7 3 2 **1**	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0			

Exploració i emplenament de 10 caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**9**) hi haurà	**1.023**	+	**59**	+	**9** = **1.091** retrocessos
- Si també omplim 5,1 (**5**) hi haurà	**453**	+	**88**	+	**6** = **547** retrocessos
- Si també omplim 9,1 (**1**) hi haurà	**296**	+	**104**	+	**6** = **406** retrocessos
- Si també omplim 2,2 (**5**) hi haurà	**66**	+	**95**	+	**6** = **167** retrocessos
- Si també omplim 4,2 (**3**) hi haurà	**80**	+	**143**	+	**0** = **223** retrocessos
- Si també omplim 6,2 (**1**) hi haurà	**59**	+	**36**	+	**6** = **101** retrocessos
- Si també omplim 8,2 (**8**) hi haurà	**10**	+	**39**	+	**0** = **49** retrocessos
- Si també omplim 1,3 (**3**) hi haurà	**13**	+	**39**	+	**0** = **52** retrocessos
- Si també omplim 5,3 (**8**) hi haurà	**4**	+	**20**	+	**0** = **24** retrocessos
- Si també omplim 9,3 (**4**) hi haurà	**7**	+	**18**	+	**5** = **30** retrocessos
Sumant retrocessos per requadres:	**2.011**	+	**641**	+	**38** = **2.690** retrocessos

Exploració i emplenament de 10 caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**8**) hi haurà	**1.134**	+	**46**	+	**28** = **1.208** retrocessos
- Si també omplim 5,4 (**4**) hi haurà	**921**	+	**28**	+	**7** = **956** retrocessos
- Si també omplim 9,4 (**3**) hi haurà	**648**	+	**33**	+	**4** = **685** retrocessos
- Si també omplim 2,5 (**4**) hi haurà	**384**	+	**22**	+	**2** = **408** retrocessos
- Si també omplim 4,5 (**2**) hi haurà	**352**	+	**176**	+	**0** = **528** retrocessos
- Si també omplim 6,5 (**3**) hi haurà	**269**	+	**14**	+	**0** = **283** retrocessos
- Si també omplim 8,5 (**9**) hi haurà	**227**	+	**6**	+	**1** = **234** retrocessos
- Si també omplim 1,6 (**2**) hi haurà	**276**	+	**1**	+	**2** = **279** retrocessos
- Si també omplim 5,6 (**9**) hi haurà	**235**	+	**0**	+	**0** = **235** retrocessos
- Si també omplim 9,6 (**5**) hi haurà	**183**	+	**1**	+	**3** = **187** retrocessos
Sumant retrocessos per requadres:	**4.629**	+	**327**	+	**47** = **5.003** retrocessos

Exploració i emplenament de 10 caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**1.134**	+	**36**	+	**14** = **1.184** retrocessos
- Si també omplim 5,7 (**6**) hi haurà	**894**	+	**39**	+	**3** = **936** retrocessos
- Si també omplim 9,7 (**2**) hi haurà	**648**	+	**14**	+	**0** = **662** retrocessos
- Si també omplim 2,8 (**6**) hi haurà	**384**	+	**5**	+	**0** = **389** retrocessos
- Si també omplim 4,8 (**1**) hi haurà	**352**	+	**0**	+	**0** = **352** retrocessos
- Si també omplim 6,8 (**2**) hi haurà	**303**	+	**3**	+	**0** = **306** retrocessos
- Si també omplim 8,8 (**7**) hi haurà	**227**	+	**3**	+	**0** = **230** retrocessos
- Si també omplim 1,9 (**1**) hi haurà	**308**	+	**0**	+	**0** = **308** retrocessos
- Si també omplim 5,9 (**7**) hi haurà	**195**	+	**0**	+	**0** = **195** retrocessos
- Si també omplim 9,9 (**8**) hi haurà	**167**	+	**0**	+	**0** = **167** retrocessos
Sumant retrocessos per requadres:	**4.612**	+	**100**	+	**17** = **4.729** retrocessos

Exploració i emplenament de les caselles 1r requadre, seguint patró habitual

- Si només omplim 1,1 (**9**) hi haurà	**1.023**	+	**59**	+	**9**	= **1.091** retrocessos
- Si també omplim 2,1 (**8**) hi haurà	**561**	+	**95**	+	**23**	= **679** retrocessos
- Si també omplim 3,1 (**7**) hi haurà	**183**	+	**31**	+	**11**	= **225** retrocessos
- Si també omplim 4,1 (**6**) hi haurà	**0**	+	**27**	+	**0**	= **27** retrocessos
- Si també omplim 5,1 (**5**) hi haurà	**0**	+	**20**	+	**2**	= **22** retrocessos
- Si també omplim 6,1 (**4**) hi haurà	**0**	+	**32**	+	**0**	= **32** retrocessos
- Si també omplim 7,1 (**3**) hi haurà	**0**	+	**19**	+	**0**	= **19** retrocessos
- Si també omplim 8,1 (**2**) hi haurà	**0**	+	**18**	+	**1**	= **19** retrocessos
i també s'omple 9,1 (**1**)						
- Si també omplim 1,2 (**6**) hi haurà	**0**	+	**40**	+	**2**	= **42** retrocessos
- Si també omplim 2,2 (**5**) hi haurà	**0**	+	**63**	+	**0**	= **63** retrocessos
- Si també omplim 3,2 (**4**) hi haurà	**0**	+	**24**	+	**0**	= **24** retrocessos
- Si també omplim 4,2 (**3**) hi haurà	**0**	+	**19**	+	**0**	= **19** retrocessos
- Si també omplim 5,2 (**2**) hi haurà	**0**	+	**12**	+	**0**	= **12** retrocessos
- Si també omplim 6,2 (**1**) hi haurà	**0**	+	**7**	+	**0**	= **7** retrocessos
- Si també omplim 7,2 (**9**) hi haurà	**0**	+	**21**	+	**0**	= **21** retrocessos
- Si també omplim 8,2 (**8**) hi haurà	**0**	+	**14**	+	**0**	= **14** retrocessos
i també s'omple 9,2 (**7**)						
- Si també omplim 1,3 (**3**) hi haurà	**0**	+	**23**	+	**0**	= **23** retrocessos
- Si també omplim 2,3 (**2**) hi haurà	**0**	+	**18**	+	**0**	= **18** retrocessos
i també s'omple 3,3 (**1**)						
- Si també omplim 4,3 (**9**) hi haurà	**0**	+	**24**	+	**0**	= **24** retrocessos
- Si també omplim 5,3 (**8**) hi haurà	**0**	+	**12**	+	**0**	= **12** retrocessos
i també s'omple 6,3 (**7**)						
- Si també omplim 7,3 (**6**) hi haurà	**0**	+	**18**	+	**0**	= **18** retrocessos
- Si també omplim 8,3 (**5**) hi haurà	**0**	+	**12**	+	**0**	= **12** retrocessos
i també s'omple 9,3 (**4**)						
Sumant retrocessos per requadres:	**1.767**	+	**608**	+	**48**	= **2.423** retrocessos

Exploració i emplenament de les caselles 2n requadre, seguint patró habitual

- Si només omplim 1,4 (**8**) hi haurà	**1.134**	+	**46**	+	**28**	= **1.208** retrocessos
- Si també omplim 2,4 (**9**) hi haurà	**1.008**	+	**43**	+	**14**	= **1.065** retrocessos
- Si també omplim 3,4 (**6**) hi haurà	**778**	+	**7**	+	**4**	= **789** retrocessos
- Si també omplim 4,4 (**7**) hi haurà	**132**	+	**4**	+	**2**	= **138** retrocessos
- Si també omplim 5,4 (**4**) hi haurà	**94**	+	**0**	+	**1**	= **95** retrocessos
- Si també omplim 6,4 (**5**) hi haurà	**80**	+	**0**	+	**2**	= **82** retrocessos
- Si també omplim 7,4 (**2**) hi haurà	**48**	+	**0**	+	**1**	= **49** retrocessos
- Si també omplim 8,4 (**1**) hi haurà	**8**	+	**0**	+	**0**	= **8** retrocessos
i també s'omple 9,4 (**3**)						
- Si també omplim 1,5 (**7**) hi haurà	**24**	+	**0**	+	**4**	= **28** retrocessos
- Si també omplim 2,5 (**4**) hi haurà	**22**	+	**0**	+	**0**	= **22** retrocessos
- Si també omplim 3,5 (**5**) hi haurà	**24**	+	**0**	+	**0**	= **24** retrocessos
- Si també omplim 4,5 (**2**) hi haurà	**12**	+	**0**	+	**4**	= **16** retrocessos
- Si també omplim 5,5 (**1**) hi haurà	**7**	+	**0**	+	**4**	= **11** retrocessos
- Si també omplim 6,5 (**3**) hi haurà	**3**	+	**0**	+	**2**	= **5** retrocessos
- Si també omplim 7,5 (**8**) hi haurà	**6**	+	**0**	+	**4**	= **10** retrocessos
- Si també omplim 8,5 (**9**) hi haurà	**4**	+	**0**	+	**2**	= **6** retrocessos
i també s'omple 9,5 (**6**)						
- Si també omplim 1,6 (**2**) hi haurà	**3**	+	**0**	+	**0**	= **3** retrocessos
- Si també omplim 2,6 (**1**) hi haurà	**4**	+	**0**	+	**0**	= **4** retrocessos
i també s'omple 3,6 (**3**)						
- Si també omplim 4,6 (**8**) hi haurà	**10**	+	**0**	+	**0**	= **10** retrocessos
- Si també omplim 5,6 (**9**) hi haurà	**10**	+	**0**	+	**0**	= **10** retrocessos
- i també s'omple 6,6 (**6**)						
- Si també omplim 7,6 (**7**) hi haurà	**13**	+	**0**	+	**0**	= **13** retrocessos
- Si també omplim 8,6 (**4**) hi haurà	**9**	+	**0**	+	**0**	= **9** retrocessos
i també s'omple 9,6 (**5**)						
Sumant retrocessos per requadres:	**3.433**	+	**100**	+	**72**	= **3.605** retrocessos

Abans de passar al requadre superior, observeu que en el central ja no es produeix cap repunt sobtat, com el de la casella 5,5 en l'emplenament homòleg de NATURMÍN.

Exploració i emplenament de les caselles 3r requadre, seguint patró habitual

- Si només omplim 1,7 (**5**) hi haurà	**1.134**	+	**36**	+	**14**	= **1.184** retrocessos
- Si també omplim 2,7 (**7**) hi haurà	**1.008**	+	**18**	+	**4**	= **1.030** retrocessos
- Si també omplim 3,7 (**9**) hi haurà	**778**	+	**4**	+	**0**	= **782** retrocessos
- Si també omplim 4,7 (**4**) hi haurà	**666**	+	**0**	+	**0**	= **666** retrocessos
- Si també omplim 5,7 (**6**) hi haurà	**570**	+	**0**	+	**0**	= **570** retrocessos
- Si també omplim 6,7 (**8**) hi haurà	**444**	+	**0**	+	**0**	= **444** retrocessos
- Si també omplim 7,7 (**1**) hi haurà	**117**	+	**0**	+	**0**	= **117** retrocessos
- Si també omplim 8,7 (**3**) hi haurà	**18**	+	**0**	+	**0**	= **18** retrocessos
i també s'omple 9,7 (**2**)						
- Si també omplim 1,8 (**4**) hi haurà	**54**	+	**0**	+	**0**	= **54** retrocessos
- Si també omplim 2,8 (**6**) hi haurà	**42**	+	**1**	+	**0**	= **43** retrocessos
- Si també omplim 3,8 (**8**) hi haurà	**35**	+	**0**	+	**0**	= **35** retrocessos
- Si també omplim 4,8 (**1**) hi haurà	**24**	+	**0**	+	**0**	= **24** retrocessos
- Si també omplim 5,8 (**3**) hi haurà	**24**	+	**467**	+	**0**	= **491** retrocessos
- Si també omplim 6,8 (**2**) hi haurà	**6**	+	**0**	+	**0**	= **6** retrocessos
- Si també omplim 7,8 (**5**) hi haurà	**18**	+	**0**	+	**0**	= **18** retrocessos
- Si també omplim 8,8 (**7**) hi haurà	**12**	+	**0**	+	**0**	= **12** retrocessos
i també s'omple 9,8 (**9**)						
- Si també omplim 1,9 (**1**) hi haurà	**18**	+	**0**	+	**0**	= **18** retrocessos
- Si també omplim 2,9 (**3**) hi haurà	**12**	+	**0**	+	**0**	= **12** retrocessos
i també s'omple 3,9 (**2**)						
- Si també omplim 4,9 (**5**) hi haurà	**18**	+	**0**	+	**0**	= **18** retrocessos
- Si també omplim 5,9 (**7**) hi haurà	**19**	+	**0**	+	**0**	= **19** retrocessos
- i també s'omple 6,9 (**9**)						
- Si també omplim 7,9 (**4**) hi haurà	**39**	+	**0**	+	**0**	= **39** retrocessos
- Si també omplim 8,9 (**6**) hi haurà	**26**	+	**0**	+	**0**	= **26** retrocessos
i també s'omple 9,9 (**8**)						
Sumant retrocessos per requadres:	**5.082**	+	**526**	+	**18**	= **5.626** retrocessos

Encara que la reducció sigui molt barroera, podríem dir (de fet, ja ho vam deixar anar poc després de presentar aquestes solucions singulars) que, als efectes de la dinàmica NATURMÍN amb què està programada l'exploració de **RE-MEMBER**, les solucions NATURMÍN i NATURMAX són dues situacions extremes, l'una a favor d'aquesta dinàmica (coincideix amb l'emplenament virtual a partir del candidat **1** a 1,1) i l'altra a contra corrent, i que, en aquest sentit, els altres SUDOKUs-SOLUCIÓ s'ubicaran en l'espai de probabilitat comprès entre aquestes. Tota aquesta retòrica agafada pels pèls anava dirigida a justificar un raonament d'escassa consistència lògica però força plausible des d'un punt de vista pràctic: si en la Solució NATURMÍN (per ser exactes, en unes poques experiències efectuades sobre aquesta solució), **RE-MEMBER** triga menys a presentar els valors admissibles per a cada nova casella quan les ja emplenades es concentren en el requadre 9×3 inferior que quan es concentren en el requadre central o en el superior, no havent-hi gran diferència entre aquests dos; si en la Solució NATURMAX (fem la mateixa precisió d'abans), **RE-MEMBER** triga menys a presentar els valors admissibles per a cada nova casella quan les ja emplenades es concentren en el requadre 9×3 inferior que quan es concentren en el requadre central, i triga menys en aquest últim cas que quan es concentren en el superior; llavors, en qualsevol sudoku en construcció en què els emplenaments es concentrin en un requadre, és més probable que **RE-MEMBER** trigui menys quan aquest requadre és l'inferior que quan és el central, i és més probable que trigui menys quan és el central que quan és el superior. D'aquí, passem a afirmar que, si les caselles ja emplenades es concentren en un únic requadre 9×3 no inferior, és més probable que **RE-MEMBER** despatxi abans la nova casella si apliquem al caseller una permutació de requadres de manera aquest se situï en posició inferior. I el pas final és afirmar que, si les caselles ja emplenades es reparteixen entre dos o tres requadres 9×3, amb un nombre diferent d'ocupades a cadascun d'ells, és més probable que **RE-MEMBER** despatxi abans la nova casella si apliquem al caseller una permutació de requadres de forma que el més poblat se situï en posició inferior, el de població intermèdia en posició central i el de població inferior o nul·la en posició superior.

Es podria argumentar, amb raó, que el crèdit que mereixi aquesta conclusió aniria en alça si els assajos amb NATURMÍN i NATURMAX no s'haguessin limitat a emplenar requadres 9×3 més o menys (els totals de cada taula reflecteixen l'emplenament de 1, 10 o 27 caselles en el requadre, però fàcilment podrem conèixer els resultats d'hipòtesis intermèdies d'emplenament obtenint sumes parcials des de l'inici de la

taula fins a qualsevol casella) i haguéssim anat més enllà estenent les ocupacions a més d'un requadre. És veritat, però, a banda de la feinada que això comportaria, si realment els nous assajos introduissin elements nous a considerar, difícilment podrien encabir-se en un dispositiu tan elemental com el que pretenem: si s'escau, transposar l'escaquer, permutar els requadres 9×3, i prou. Perquè, com sol passar quan s'està a punt d'assolir un difícil equilibri, cal vigilar que l'afany de més precisió no acabi llastant el procés (allò de *lo mejor es enemigo de lo bueno*).

Més incisiva seria una crítica que anés directament a desmentir la major, posant en qüestió la "normalització" aplicada a la taula d'emplenament total del requadre central de NATURMÍN. Perquè, ¿qui ens autoritzava a falsejar les dades de sortida, escapçant el pic de retrocessos corresponent a l'emplenament de la casella 5,5? Tant si aquell repunt posava en evidència la limitada capacitat dels dispositius optimitzadors introduïts a **RE-MEMBER** com l'existència d'algun fenòmen alentidor no descobert (l'únic que podem assegurar, perquè l'autor ho ha comprovat, és que la plena disponibilitat d'**EXPLORA-3*9** en **RE-MEMBER** no hauria servit absolutament de res en aquell cas), no era el producte d'algun factor aliè al procés d'emplenament sinó una dada tan autèntica i significativa com les demés. En aquest sentit, hem fet trampa per poder dir que el total de retrocessos en aquest requadre central, **10.534 - 3.000 = 7.534**, era si fa no fa igual als **7.949** retrocessos del superior, i que el sil·logisme imperfecte d'abans s'hagués hagut de formular així: si en la Solució NATURMÍN (per ser exactes, en unes poques experiències efectuades sobre aquesta solució), **RE-MEMBER** triga menys a presentar els valors admissibles per a cada nova casella quan les ja emplenades es concentren en el requadre 9×3 inferior que quan es concentren en el requadre superior, i triga menys en aquest últim cas que quan es concentren en el central; si en la Solució NATURMAX (fem la mateixa precisió d'abans), **RE-MEMBER** triga menys a presentar els valors admissibles per a cada nova casella quan les ja emplenades es concentren en el requadre 9×3 inferior que quan es concentren en el requadre central, i triga menys en aquest últim cas que quan es concentren en el superior; llavors, en qualsevol sudoku en construcció on els emplenaments es concentrin en un requadre, és més probable que **RE-MEMBER** trigui menys quan aquest requadre és l'inferior que quan és el central o superior. D'aquí, passem a afirmar que, si les caselles ja emplenades es concentren en un únic requadre 9×3 no inferior, és més probable que **RE-MEMBER** despatxi abans la nova casella si apliquem al caseller una permutació de requadres de manera aquest se situï en posició inferior. I el pas final és afirmar que, si les caselles ja emplenades es reparteixen entre dos o tres requadres 9×3, amb un nombre diferent d'ocupades a cadascun d'ells, és més probable que **RE-MEMBER** despatxi abans la nova casella si apliquem al caseller una permutació de requadres de forma que el més poblat se situï en posició inferior, resultant irrellevant com quedin situats els altres dos requadres.

Doncs bé, tant si el lector opta per la primera formulació com per la segona, com que en algun lloc haurem de posar el requadre mitjanament poblat i el menys poblat de tots, seguirem la primera pauta. Fins i tot si algun lector recalcitrant té la dèria de situar a dalt el requadre mitjanament poblat i al mig el menys poblat de tots, podrà fer-ho introduint el petit canvi indicat en un comentari al codi que de seguida presentem. Abans però, convé indicar que el pic de **3.274** retrocessos es tradueix en una espera d'uns 20 segons pel que fa al candidat **7** i, si mantenim la versió que permet treure per pantalla el desglossament per candidats, veurem:

```
     1<     0 +     0 + 0 =     0
     2<     0 +     0 + 0 =     0
     4<     0 +     0 + 0 =     0
     7> 1.612 + 1.586 + 0 = 3.198
     9>    74 +     2 + 0 =    76
     T> 1.686 + 1.588 + 0 = 3.274
```

Aquests 20 segons poden semblar inadmissibles, però són tot un triomf si pensem que abans d'optimitzar **RE-MEMBER** alguns temps d'espera es podien mesurar en hores. A més, a conseqüència de la permutació de requadres 9×3, aquest requadre central passarà a situar-se en posició inferior i l'aparició dels dos candidats aptes serà pràcticament instantània, amb el desglossament:

```
     1<  0 +  0 + 0 =  0
     2<  0 +  0 + 0 =  0
     4<  0 +  0 + 0 =  0
     7>  9 +  7 + 0 = 16
     9>  9 +  4 + 0 = 13
     T> 18 + 11 + 0 = 29
```

És clar que, si ens adonem que el conflicte que provocava el repunt amb el candidat **7** a 5,5 era la impossibilitat de completar la 5ª fila, perquè ja no hi quedava lloc pel **9** i no hi havia més remei que retrocedir fins a la 2ª fila per poder-hi permutar el **8** amb el **9**, podem pensar una situació d'emplenament com la de la dreta, en què el problema quasi és idèntic però el requadre inferior té 14 caselles plenes com el central (comptant-hi la casella 5,5 activada), cosa que inhibirà el dispositiu que permuta requadres. La pausa ha baixat a 10 segons (**1.827** retrocessos) només perquè 4,2 i 6,3 són plenes. Tant de bo no superem mai aquest valor!

```
0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0
3 6 5   8 7 0   0 0 0
2 1 4   3 6 5   8 9 7

0 0 0   0 0 3   0 0 0
4 5 6   7 0 0   0 0 0
1 2 3   4 5 6   7 8 9
```

Ja ens hem esplaiat prou justificant la pertinència d'aquest dispositiu i ara toca descriure'n el funcionament, cosa que aprofitarem per reagrupar el codi que gira a l'entorn d'**OMPLE-SUDOKU**, amb què havíem obert la primera entrega d'aquest capítol (*Etapa prèvia: crear una solució, I*) i n'hem seguit l'evolució, però sols de forma fragmentària. En acabat, n'aclarirem aquells aspectes que necessitin explicacions.

```
(defun SUB-X*Y (X*Y P1 P2 / J K L LL)
  (setq K -1)
  (foreach Y X*Y
    (setq K (1+ K) J -1)
    (if (not (or (< K (cadr P1)) (> K (cadr P2))))
        (progn
          (foreach X Y
            (setq J (1+ J))
            (if (not (or (< J (car P1)) (> J (car P2)))) (setq L (cons X L))))
          (setq LL (cons (reverse L) LL) L ()))))
  (reverse LL))

(defun ACT-LLL (X Y A-LLL / X/3 Y/3 LL L J I)
  (setq X/3 (/ X 3) Y/3 (/ Y 3) J -1)
  (foreach F A-LLL
    (setq L () I -1 J (1+ J))
    (foreach E F
      (setq I (1+ I)
            L (cons (if (= J Y)
                        (if (= I X) () (subst () A-N E))
                        (if (= I X)
                            (subst () A-N E)
                            (if (and (= (/ I 3) X/3) (= (/ J 3) Y/3))
                                (subst () A-N E) E)))
                    L)))
    (setq LL (cons (reverse L) LL)))
  (reverse LL))

(defun PLUS-N->P ()
  (setq A-N N A-P K PLUS T)
  (if REAL (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))))

(defun ACTUALITZA-PLUS (/ 3*3 LL3 M N X1 Y1)
  (mapcar '(lambda (F-9*9 F-LLL)
             (if (not PLUS)
                 (mapcar '(lambda (E-9*9 E-LLL)
                            (if (and (not PLUS) E-LLL)
                                (progn
                                  (setq K 0)
                                  (foreach M L1A9
                                    (if (and (< K 2) (member M E-LLL))
                                        (setq N M K (1+ K))))
                                  (if (= K 1)
                                      (progn (setq K E-9*9) (PLUS-N->P))))))
                         F-9*9 F-LLL)))
          A-9*9 A-LLL)
  (if (not PLUS)
      (foreach Y1 '(0 3 6)
        (foreach X1 '(0 3 6)
          (if (not PLUS)
```

```
              (progn
                (setq 3*3 (SUB-X*Y A-9*9 (list X1 Y1) (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL (list X1 Y1) (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0)
                        (mapcar '(lambda (F-3*3 F-LL3)
                                   (mapcar '(lambda (E-3*3 E-LL3)
                                              (if (member N E-LL3)
                                                  (setq K (1+ K) M E-3*3)))
                                           F-3*3 F-LL3))
                                3*3 LL3)
                        (if (= K 1) (progn (setq K M) (PLUS-N->P))))))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L LL PLUS)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E) LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (ACTUALITZA-PLUS)
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))

(defun EXPL-SUBCONJ (NUM / LA J K NO)
  (if NUM
     (foreach M (reverse L1A9)
       (setq NO T)
       (foreach E L
         (if (and NO (member M E)) (setq NO () LA (cons (list M) LA)))))
     (repeat (setq K (length L)) (setq K (1- K) LA (cons (list K) LA))))
  (setq LL LA K 1)
  (repeat (1- (length LL))
    (if (not FORA)
        (progn
          (setq K (1+ K) LA LL LL ())
          (repeat (1- (length LA))
            (setq I (car LA) LA (cdr LA) J (reverse (cdr (reverse I))))
            (foreach E LA
              (if (equal (reverse (cdr (reverse E))) J)
                  (setq LL (cons (append I (list (last E))) LL)))))
          (setq LL (reverse LL))
          (foreach F LL
            (if (not FORA)
                (progn
                  (if NUM
                      (progn
                        (setq I 0)
                        (foreach M L
                          (setq NO T)
                          (foreach E F
                            (if (and NO (member E M)) (setq I (1+ I) NO ()))))))
                      (progn
                        (setq LA ())
                        (foreach E F
                          (foreach M (nth E L)
                            (if (not (member M LA)) (setq LA (cons M LA)))))
                        (setq I (length LA))))
                  (if (< I K) (setq FORA T)))))))))

(defun EXPLORA-3*9 (/ FORA R-9 R-L L LL I M)
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y (car 2R) (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y (cadr 2R) (list X Y) (list (+ X 2) (+ Y 2))) L ())
```

```
;           ********************************************************************
            (foreach F R-9 (foreach E F (if (atom E) (setq L (cons E L)))))     ;*
            (foreach F R-L                                                      ;*
              (if (not FORA)                                                    ;*
                  (foreach E F                                                  ;*
                    (if E (foreach M E                                          ;*
                            (if (and M (not (member M L)))                      ;*
                                (setq L (cons M L))))))))                       ;*
            (if (and (not FORA) (< (length L) 9)) (setq FORA T))                ;*
;           ********************************************************************
            (if (and INI (not FORA))
                (progn
                  (setq L ())
                  (foreach F R-L
                    (foreach E F
                      (if E (progn
                              (setq I ())
                              (foreach M E (if M (setq I (cons M I))))
                              (setq L (cons I L))))))
                  (EXPL-SUBCONJ ())))))
  FORA)

(defun COMPLET-9*9 (9*9 / COMP)
  (setq COMP T)
  (foreach L 9*9
    (if COMP
        (foreach E L
          (if (and COMP (listp E)) (setq COMP () R-P (if POST E))))))
  COMP)

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                INI (EXPLORA-3*9) R-9*9 (car 2R) R-LLL (cadr 2R)))
  (if (not (or INI OK))
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if (not OK) (setq OK (EXPLORA-3*9)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)

(defun TECLAT ()
  (command "ESPACIOP"
           "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.2,-0.2" "-0.1,-0.1"
           "COPIA" "LT" "" "-1.0419,0.5919" ""
           "EDITPOL" (setq CURSOR (ssget "_L")) "G" 0.006 ""
           "MATRIZ" "-0.1,-0.1" "" "R" 3 3 0.15 0.15
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "COLOR" G
           "SOMBREA" "S" "C" "-0.22,-0.53" "0.22,0.53" ""
           "COLOR" "PORCAPA")
  (MASCARA L1A9)
  (if (and (= ABC "A") (not RESP) (not 1-2))
      (command "ESPACIOM" "CVPORT" 2 "CAPA" "D" "NORM-1" "")))

(defun M->P () (list (+ (* 0.1105 X) -1.1919) (+ (* 0.1105 Y) -0.4419)))

(defun ELEMENT (LL) (nth X (nth Y LL)))
```

```
(defun TECLA-M ()
  (while (not (and (= (car (setq GR (grread))) 3)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (equal (list PX PY) '(4 4) 4.48)
                   (or (< (- PX (setq X (fix PX))) 0.48)
                       (< (- (setq X (fix (1+ PX))) PX) 0.48))
                   (or (< (- PY (setq Y (fix PY))) 0.48)
                       (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                   (ELEMENT LLL))))
  (setq P (list X Y)))

(defun TECLA-P (VARS / FI)
  (while (not (and (or (= (car (setq GR (grread))) 3)
                       (setq FI (and VARS (> (strlen MSG) 51)
                                     (or (equal GR '(2 13)) (equal GR '(2 32))))))
                   (or FI (equal (setq GR (cadr GR)) GR (list (car GR) (cadr GR)))
                                 '(0 0) 0.2))
                   (or FI (setq N (cond ((equal GR '(-0.15  0.15) 0.05) 1)
                                        ((equal GR '( 0     0.15) 0.05) 2)
                                        ((equal GR '( 0.15  0.15) 0.05) 3)
                                        ((equal GR '(-0.15  0   ) 0.05) 4)
                                        ((equal GR '( 0     0   ) 0.05) 5)
                                        ((equal GR '( 0.15  0   ) 0.05) 6)
                                        ((equal GR '(-0.15 -0.15) 0.05) 7)
                                        ((equal GR '( 0    -0.15) 0.05) 8)
                                        ((equal GR '( 0.15 -0.15) 0.05) 9))))
                   (or VARS (= ABC "B") (member N LL))))))

(defun INI-COORDS ()
  (setq Y -1 PY ())
  (reverse (repeat 9 (setq Y (1+ Y) X -1 PX ()
                           PY (cons (reverse (repeat 9 (setq X (1+ X)
                                                             PX (cons (list X Y)
                                                                      PX))))
                                    PY)))))

(defun INI-LLL ()
  (setq LL () LLL ())
  (repeat 9 (setq LL (cons L1A9 LL)))
  (repeat 9 (setq LLL (cons LL LLL))))

(defun N-3*3 (9*9 / 3*3 *3 I)
  (setq K -1)
  (repeat 3 ; 3 requadres 9*3
    (setq *3 ())
    (repeat 3 ; 1 requadre 9*3 = 3 files
      (setq J -1 K (1+ K))
      (repeat 3 ; 1 fila = 3 tercets
        (setq N 0)
        (repeat 3 ; 1 tercet = 3 caselles
          (setq J (1+ J) I (nth J (nth K 9*9)))
          (if (or (atom I) (equal I P)) (setq N (1+ N))))
        (setq *3 (cons N *3))))
    (setq *3 (list (+ (nth 2 *3) (nth 5 *3) (nth 8 *3))
                   (+ (nth 1 *3) (nth 4 *3) (nth 7 *3))
                   (+ (nth 0 *3) (nth 3 *3) (nth 6 *3))) 3*3 (cons *3 3*3)))
  (setq 3*3 (reverse 3*3)))

(defun TRANSPOSA-HO ()
  (setq K (TRANSPOSAR **9*9**) LLL (TRANSPOSAR LLL) P (reverse P) 9**9 ())
  (foreach EE K
    (setq 0-L ())
    (foreach E EE (setq 0-L (cons (if (atom E) E (reverse E)) 0-L)))
    (setq 9**9 (cons (reverse 0-L) 9**9)))
  (setq 9**9 (reverse 9**9) **9*9** 9**9 K X X Y Y K))

(defun 1+2+3 (LL) (mapcar '(lambda (L) (eval (cons '+ L))) LL))
```

```
(defun F=3 (9*9)
  (setq 9**9 ())
  (foreach J JJ
    (setq K (1- (* 3 J)))
    (repeat 3 (setq K (1+ K) 9**9 (cons (nth K 9*9) 9**9))))
  (reverse 9**9))

(defun TREU-RETOL () (repeat 4 (command "BORRA" "LT" "")))

(defun MASCARA (L / INI OK Ñ JJ 9**9 0-**9*9** 0-LLL 0-L)
  (setq LL () SS (ssadd))
  (if POST
      (progn
        (setq N (N-3*3 **9*9**) Ñ (TRANSPOSAR N) N (1+2+3 N) Ñ (1+2+3 Ñ))
        (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
            (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
        (if (and (>= (car N) (cadr N)) (>= (cadr N) (caddr N)))
            (setq 0-**9*9** **9*9** 0-LLL LLL)
            (progn
              (setq JJ (if (and (<= (car N) (cadr N)) (<= (cadr N) (caddr N)))
                           '(2 1 0)
                           (if (>= (car N) (caddr N))
                               (if (> (cadr N) (caddr N)) '(1 0 2) '(0 2 1))
                               (if (> (cadr N) (caddr N)) '(1 2 0) '(2 0 1))))
                    0-LLL (F=3 **9*9**) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**) 0-LLL (F=3 LLL)
                    P (list X (+ (rem Y 3)
                                 (* 3 (- 3 (length (member (/ Y 3) JJ)))))))))
        (if (< (car Ñ) (caddr Ñ))
            (progn
              (setq K ())
              (foreach EE (reverse 0-**9*9**)
                (setq 0-L ())
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                  0-L)))
                (setq K (cons 0-L K)))
              (setq 0-**9*9** K K ())
              (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
              (setq 0-LLL K P (list (- 8 (car P)) (cadr P)))))))
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "CLIC sobre"
                      "el nombre..."))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (if POST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         ((= N 2)     '(0 0.15))
                         ((= N 3)  '(0.15 0.15))
                         ((= N 4) '(-0.15 0))
                         ((= N 5)     '(0 0))
                         ((= N 6)  '(0.15 0))
                         ((= N 7) '(-0.15 -0.15))
                         ((= N 8)     '(0 -0.15))
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if POST (TREU-RETOL))))
  (if POST
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))
```

```
(defun OMPLE-SUDOKU (/ CURSOR P-CURS SS C->F)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq **9*9** (INI-COORDS) P-CURS '(-1.1919 0.4419) POST T)
        (INI-LLL)
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP" "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (TECLA-P ())
               (command "ESPACIOM" "CVPORT" 2
                        "CAPA" "D" "NORM-1" "" "TEXTO" "MC" P 0.5 0 (itoa N)
                        "CVPORT" 3
                        "CAPA" "D" "VAR-1"  "" "TEXTO" "MC" P 0.5 0 (itoa N)
                        "CVPORT" 2)
               (setq LLL (ACTUALITZA **9*9** N P LLL T)
                     **9*9** (car LLL) LLL (cadr LLL)))
        (command "ESPACIOP")
        (setq POST ()))
      (progn
        (prompt "\n.\n.\nCLIC sobre el nombre que va a la casella marcada: ")
        (setq K 9)
        (repeat 9
          (setq L () J -1 K (1- K))
          (repeat 9
            (setq J (1+ J))
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.110465,0" ""
                     "ESPACIOM" "CVPORT" 2
                     "CAPA" "D" "NORM-1" "" "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "CVPORT" 3
                     "CAPA" "D" "VAR-1"  "" "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "ESPACIOP")
            (setq L (append L (list N))))
          (command "DESPLAZA" CURSOR "" "-0.9942,-0.1105" "")
          (setq **9*9** (cons L **9*9**)))))
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))
```

En relació al dispositiu que, transposant i/o permutant ****9*9**** i **LLL**, els deixa com arguments **O-**9*9**** i **O-LLL** òptims per a **RE-MEMBER**, ens limitarem a explicar-ne succintament la mecánica. La funció **N-3*3** prepara una llista **N** de 3 requadres 9×3, cada un representat per una llista de 3 nombres: les caselles ocupades pels seus requadres 3×3 (l'actual hi compta com una més). Mitjançant la funció **1+2+3**, construïm les llistes **N** i **Ñ** de 3 valors, que representen respectivament el nombre de caselles plenes dels requadres 9×3 i 9×3, i que ens permetrà veure quina de les dues particions en subconjunts de 27 caselles té un element més poblat: si un dels requadres verticals és qui té més caselles emplenades, transformarem ****9*9**** i **LLL** amb **TRANSPOSA-HO** (que, com havíem dit a l'inici del capítol, obté les transposades d'aquestes pseudomatrius, permutant les coordenades dels elements del primer que corresponen a caselles buides, i també les de **P**), fet registrat en l'activació del senyal **C->F**; altrament seguirem treballant amb les originals. La funció **F=3**, usada per **CAN->VAR** en el capítol *Tornem a començar: de la solución al problema*, permuta els requadres 9×3 al dictat de la llista **JJ** (que indica la localització en què cal deixar el primer, segon i tercer requadre) i aquí ho farà de manera que s'ordenin de baix a dalt en el sentit d'ocupació decreixent; naturalment, les coordenades Y a ****9*9*** i a **P** hauran de ser actualitzades.

Renunciarem a afinar aquesta ordenació aplicant el mateix criteri regulador a les files de cada requadre 9×3, perquè no és segur que la complicació sigui rendible, però si que intervindrem mínimament en direcció transversal permutant el requadre 3×9 esquerre amb el dret (el del mig no el tocarem), quan el segon estigui més poblat que el primer, bàsicament per evitar que un requadre 3×9 buit a l'esquerra pugui enterbolir la millora deguda a l'encertada disposició dels requadres 9×3.

Qui no tingui paciència ja no haurà arribat aquí. Els lectors pacients, potser sí. Però la resta de subcapítol sols és apte per als qui, ultra tenir una paciència de Job, siguin, com l'autor, badocs impenitents sense cap sentit pràctic de la vida. Perquè, més que un epíleg, el que ve ara és una divagació originalment inserida a la pàgina 181 (aquest mateix subcapítol), que encara dificultava més el seguiment d'un discurs massa farragós i que, en lloc de ser eliminada sense contemplacions, l'autor va decidir traslladar al final, no fos cas que algun lector perseverant en pogués treure profit. Heus-la aquí.

Abans d'adaptar el programa en la forma que ja coneixeu i que ara repetim, per fer possible la sortida d'informació relativa als retrocessos,

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R))
                    (if (not OK)
                        (progn
                          (setq *NIV* (cadr R-P))
                          (cond ((< *NIV* 3) (setq *REQ-1* (1+ *REQ-1*)))
                                ((< *NIV* 6) (setq *REQ-2* (1+ *REQ-2*)))
                                (     T      (setq *REQ-3* (1+ *REQ-3*))))))))))))
  OK)
```

se n'havia ideat una altra de molt semblant,

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R)))))
            (if (not OK)
                (progn
                  (setq *NIV* (cadr R-P))
                  (cond ((< *NIV* 3) (setq *REQ-1* (1+ *REQ-1*)))
                        ((< *NIV* 6) (setq *REQ-2* (1+ *REQ-2*)))
                        (     T      (setq *REQ-3* (1+ *REQ-3*))))))))))
  OK)
```

En sabríeu apreciar la diferencia funcional? No és molt evident, perquè les dues, amb un codi estructurat gairebé igual, sembla que portin al mateix resultat: fer que apareguin en pantalla, desglossats per requadres 9×3, els retrocessos de cada candidat en la seva trajectòria d'emplenament virtual de l'escaquer, impulsada per **RE-MEMBER**. Això és cert quan el candidat rep de la funció un veredicte favorable, però quan sigui descartat i no arribi a aparèixer en el caseller 3×3 controlat per **MASCARA**, malgrat ser compatible en primera instància, a banda de sortir precedits pel caràcter "<" i no per ">", si la versió utilitzada és la segona, el recompte superarà en una unitat el resultat de la primera. La disparitat de respostes no té res d'estrany perquè, si en l'exploració es retrocedeix a l'última casella lliure (retorn al nivell de recursió precedent) en esgotar-se els candidats de l'actual amb un resultat **(not OK)**, la diferència entre ambdues versions rau en la casella

des d'on es fa l'anotació del nou retrocés: la primera versió la fa després (és a dir, a la casella on acaba) i la segona la fa abans (a la casella on s'inicia o es vol iniciar el retrocés, perquè pot ser que enrera ja no quedin caselles lliures). Bé, pero ¿que no és el mateix comptar el pas enrera abans o després de donar-lo? Sí, si l'exploració acaba amb un pas endavant (arriba a l'última casella lliure, li assigna l'únic valor compatible i s'activa el senyal **OK** que clou l'exploració), però quan acabem amb un pas enrera (que retorna a la primera casella lliure, a la qual ja no li queden més candidats, cloent també l'exploració però amb el senyal **(not OK)**) els còmputs no coincidiran: si la primera versió compta N retrocessos, la segona comptarà N+1 intencions de retrocés, l'última de les quals ja no s'haurà pogut materialitzar.

Hem portat a col·lació aquesta segona versió perquè ens va millor per mostrar les relacions numèriques que es donen quan emplenem l'escaquer 9×9 seguint el criteri NATURMÍN, reflectides en els llistats: desglossats per requadres 9×3, el nombre de retrocessos corresponent al candidat vàlidat més baix (el que adoptarem, fidels al criteri esmentat), sumant-hi si s'escau els corresponents als candidats més baixos desestimats, és igual al corresponent al candidat validat més baix de la casella precedent (l'última assignació). Cal afegir que, perquè aquest enunciat funcioni sense excepcions, tots els emplenaments han de ser correlatius, raó per la qual caldrà desactivar el dispositu tapaforats (el recurs més expeditiu és posar fora de servei els dos accesos a **ACTUALITZA-PLUS** que té **ACTUALITZA**, posant-hi sengles ";" al davant). D'aquesta manera tindrem llistats de totes les caselles i no hi haurà solucions de continuïtat en la informació: així s'ha fet per obtenir els 81 llistats que, agrupats per files, trobareu més endavant, aplegats en tres pàgines; per tal que el lector reconegui quins llistats no haurien aparegut en condicions ordinàries, en haver-se emplenat automàticament la casella que representen, els valors de l'última línia no van en negreta (parlem de la línia de Totals "T> ...", del tot prescindible en els casos de candidat únic, que són 28 dels 33 esmentats). Però abans d'aquestes tres pàgines, entre la present i les quatre següents, en teniu 81 més amb els retrocessos de les trajectòries d'exploració corresponents als nou candidats de cada casella, que suposarem activada amb l'escaquer 9×9 buit, on els relatius als emplenaments NATURMÍN hi figuren subratllats:

1ª fila

1> 246 + 49 + 15 = 310	1> 246 + 49 + 15 = 310	1> 246 + 61 + 15 = 322
2> 246 + 49 + 15 = 310	2> 246 + 49 + 15 = 310	2> 246 + 61 + 15 = 322
3> 246 + 61 + 15 = 322	3> 246 + 61 + 15 = 322	3> 246 + 49 + 15 = 310
4> 246 + 62 + 87 = 395	4> 246 + 49 + 48 = 343	4> 246 + 49 + 15 = 310
5> 246 + 36 + 95 = 377	5> 246 + 24 + 40 = 310	5> 246 + 36 + 15 = 297
6> 246 + 24 + 5 = 275	6> 246 + 24 + 40 = 310	6> 246 + 36 + 15 = 297
7> 184 + 37 + 95 = 316	7> 184 + 36 + 76 = 296	7> 184 + 36 + 15 = 235
8> 169 + 65 + 95 = 329	8> 169 + 64 + 98 = 331	8> 169 + 67 + 15 = 251
9> 164 + 64 + 95 = 323	9> 164 + 64 + 98 = 326	9> 164 + 67 + 15 = 246
T> **1993** + **447** + **517** = **2957**	T> **1993** + **420** + **445** = **2858**	T> **1993** + **462** + **135** = **2590**

1> 246 + 49 + 15 = 310	1> 246 + 11 + 129 = 386	1> 246 + 62 + 43 = 351
2> 246 + 49 + 15 = 310	2> 246 + 11 + 105 = 362	2> 246 + 62 + 17 = 325
3> 246 + 49 + 15 = 310	3> 246 + 19 + 105 = 370	3> 246 + 63 + 15 = 324
4> 246 + 49 + 15 = 310	4> 246 + 19 + 13 = 278	4> 246 + 63 + 15 = 324
5> 246 + 19 + 13 = 278	5> 246 + 49 + 15 = 310	5> 246 + 49 + 15 = 310
6> 246 + 62 + 15 = 323	6> 246 + 49 + 15 = 310	6> 246 + 49 + 15 = 310
7> 36 + 20 + 37 = 93	7> 36 + 27 + 43 = 106	7> 36 + 147 + 46 = 229
8> 36 + 27 + 37 = 100	8> 36 + 24 + 36 = 96	8> 36 + 150 + 32 = 218
9> 36 + 20 + 17 = 73	9> 36 + 9 + 54 = 99	9> 36 + 150 + 32 = 218
T> **1584** + **344** + **179** = **2107**	T> **1584** + **218** + **515** = **2317**	T> **1584** + **795** + **230** = **2609**

1> 36 + 78 + 15 = 129	1> 36 + 42 + 51 = 129	1> 36 + 41 + 24 = 101
2> 36 + 78 + 15 = 129	2> 36 + 42 + 51 = 129	2> 36 + 41 + 46 = 123
3> 36 + 78 + 4 = 118	3> 36 + 38 + 31 = 105	3> 36 + 37 + 38 = 111
4> 36 + 79 + 48 = 163	4> 36 + 39 + 31 = 106	4> 36 + 38 + 38 = 112
5> 36 + 137 + 33 = 206	5> 36 + 146 + 36 = 218	5> 36 + 148 + 50 = 234
6> 36 + 147 + 46 = 229	6> 36 + 148 + 59 = 243	6> 36 + 148 + 50 = 234
7> 246 + 49 + 15 = 310	7> 246 + 49 + 15 = 310	7> 246 + 48 + 15 = 309
8> 246 + 49 + 15 = 310	8> 246 + 49 + 15 = 310	8> 246 + 48 + 15 = 309
9> 246 + 48 + 15 = 309	9> 246 + 48 + 15 = 309	9> 246 + 49 + 15 = 310
T> **954** + **743** + **206** = **1903**	T> **954** + **601** + **304** = **1859**	T> **954** + **598** + **291** = **1843**

2ª fila

1>	246	+ 49	+ 15	= 310	1>	246	+ 62	+ 89	= 397	1>	246	+ 49 + 127 = 422
2>	246	+ 49	+ 15	= 310	2>	246	+ 62	+ 89	= 397	2>	246	+ 49 + 55 = 350
3>	246	+ 49	+ 15	= 310	3>	246	+ 62	+ 67	= 375	3>	246	+ 12 + 14 = 272
4>	246	+ 49	+ 15	= 310	4>	246	+ 62	+ 67	= 375	4>	246	+ 11 + 14 = 271
5>	246	+ 62	+ 67	= 375	5>	246	+ 49	+ 15	= 310	5>	246	+ 49 + 15 = 310
6>	246	+ 49	+ 91	= 386	6>	246	+ 49	+ 15	= 310	6>	246	+ 49 + 15 = 310
7>	36	+ 150	+ 119	= 305	7>	36	+ 150	+ 43	= 229	7>	36	+ 160 + 50 = 246
8>	36	+ 149	+ 78	= 263	8>	36	+ 150	+ 42	= 228	8>	36	+ 160 + 50 = 246
9>	36	+ 149	+ 78	= 263	9>	36	+ 149	+ 50	= 235	9>	36	+ 160 + 50 = 246
T>	**1584**	+ **755**	+ **493**	= **2832**	T>	**1584**	+ **795**	+ **477**	= **2856**	T>	**1584**	+ **699** + **390** = **2673**

1>	98	+ 160	+ 50	= 308	1>	98	+ 56	+ 12	= 166	1>	98	+ 58 + 11 = 167
2>	98	+ 160	+ 50	= 308	2>	98	+ 56	+ 12	= 166	2>	98	+ 58 + 12 = 168
3>	98	+ 41	+ 24	= 163	3>	98	+ 62	+ 11	= 171	3>	98	+ 60 + 48 = 206
4>	60	+ 36	+ 15	= 111	4>	60	+ 60	+ 22	= 142	4>	60	+ 62 + 15 = 137
5>	60	+ 36	+ 15	= 111	5>	60	+ 25	+ 102	= 187	5>	60	+ 45 + 13 = 118
6>	60	+ 36	+ 15	= 111	6>	60	+ 26	+ 7	= 93	6>	60	+ 45 + 13 = 118
7>	60	+ 49	+ 15	= 124	7>	60	+ 49	+ 15	= 124	7>	60	+ 47 + 15 = 122
8>	60	+ 49	+ 15	= 124	8>	60	+ 49	+ 15	= 124	8>	60	+ 47 + 15 = 122
9>	60	+ 41	+ 15	= 116	9>	60	+ 47	+ 15	= 122	9>	60	+ 49 + 15 = 124
T>	**654**	+ **608**	+ **214**	= **1476**	T>	**654**	+ **430**	+ **211**	= **1295**	T>	**654**	+ **471** + **157** = **1282**

1>	62	+ 49	+ 15	= 126	1>	62	+ 56	+ 15	= 133	1>	62	+ 56 + 15 = 133
2>	62	+ 56	+ 15	= 133	2>	62	+ 49	+ 15	= 126	2>	62	+ 49 + 28 = 139
3>	62	+ 68	+ 26	= 156	3>	62	+ 49	+ 28	= 139	3>	62	+ 49 + 15 = 126
4>	16	+ 159	+ 47	= 222	4>	16	+ 159	+ 26	= 201	4>	16	+ 161 + 26 = 203
5>	16	+ 180	+ 9	= 205	5>	16	+ 159	+ 26	= 201	5>	16	+ 160 + 50 = 226
6>	16	+ 180	+ 9	= 205	6>	16	+ 159	+ 26	= 201	6>	16	+ 160 + 50 = 226
7>	21	+ 169	+ 11	= 201	7>	21	+ 147	+ 14	= 182	7>	21	+ 147 + 46 = 214
8>	26	+ 169	+ 44	= 239	8>	26	+ 147	+ 15	= 188	8>	26	+ 150 + 32 = 208
9>	31	+ 168	+ 44	= 243	9>	31	+ 147	+ 15	= 193	9>	31	+ 150 + 32 = 213
T>	**312**	+ **1198**	+ **220**	= **1730**	T>	**312**	+ **1072**	+ **180**	= **1564**	T>	**312**	+ **1082** + **294** = **1688**

3ª fila

1>	36	+ 159	+ 47	= 242	1>	36	+ 155	+ 85	= 276	1>	36	+ 148 + 53 = 237
2>	36	+ 159	+ 47	= 242	2>	36	+ 155	+ 85	= 276	2>	36	+ 148 + 55 = 239
3>	36	+ 159	+ 37	= 232	3>	36	+ 155	+ 85	= 276	3>	36	+ 38 + 21 = 95
4>	36	+ 161	+ 26	= 223	4>	36	+ 157	+ 72	= 265	4>	36	+ 38 + 21 = 95
5>	36	+ 160	+ 50	= 246	5>	36	+ 27	+ 22	= 85	5>	36	+ 149 + 22 = 207
6>	36	+ 160	+ 50	= 246	6>	36	+ 27	+ 22	= 85	6>	36	+ 149 + 22 = 207
7>	246	+ 49	+ 15	= 310	7>	246	+ 49	+ 15	= 310	7>	246	+ 48 + 15 = 309
8>	246	+ 49	+ 15	= 310	8>	246	+ 49	+ 15	= 310	8>	246	+ 48 + 15 = 309
9>	246	+ 48	+ 15	= 309	9>	246	+ 48	+ 15	= 309	9>	246	+ 49 + 15 = 310
T>	**954**	+ **1104**	+ **302**	= **2360**	T>	**954**	+ **822**	+ **416**	= **2192**	T>	**954**	+ **815** + **239** = **2008**

1>	130	+ 49	+ 15	= 194	1>	130	+ 56	+ 15	= 201	1>	130	+ 54 + 15 = 199
2>	130	+ 56	+ 15	= 201	2>	130	+ 49	+ 15	= 194	2>	130	+ 49 + 9 = 188
3>	130	+ 12	+ 41	= 183	3>	130	+ 49	+ 9	= 188	3>	130	+ 49 + 15 = 194
4>	36	+ 78	+ 28	= 142	4>	36	+ 78	+ 28	= 142	4>	36	+ 79 + 48 = 163
5>	36	+ 42	+ 81	= 159	5>	36	+ 25	+ 53	= 114	5>	36	+ 137 + 33 = 206
6>	36	+ 41	+ 59	= 136	6>	36	+ 27	+ 36	= 99	6>	36	+ 147 + 46 = 229
7>	230	+ 58	+ 41	= 329	7>	230	+ 158	+ 25	= 413	7>	230	+ 160 + 50 = 440
8>	349	+ 61	+ 41	= 451	8>	349	+ 158	+ 25	= 532	8>	349	+ 160 + 50 = 559
9>	468	+ 53	+ 48	= 569	9>	468	+ 142	+ 26	= 636	9>	468	+ 160 + 50 = 678
T>	**1545**	+ **450**	+ **369**	= **2364**	T>	**1545**	+ **742**	+ **232**	= **2519**	T>	**1545**	+ **995** + **316** = **2856**

1>	361	+ 160	+ 50	= 571	1>	361	+ 158	+ 11	= 530	1>	361	+ 158 + 11 = 530
2>	408	+ 160	+ 50	= 618	2>	408	+ 160	+ 28	= 596	2>	408	+ 159 + 50 = 617
3>	455	+ 41	+ 24	= 520	3>	455	+ 38	+ 29	= 522	3>	455	+ 37 + 29 = 521
4>	246	+ 49	+ 15	= 310	4>	246	+ 65	+ 36	= 347	4>	246	+ 68 + 36 = 350
5>	246	+ 65	+ 36	= 347	5>	246	+ 49	+ 15	= 310	5>	246	+ 49 + 15 = 310
6>	246	+ 64	+ 15	= 325	6>	246	+ 49	+ 15	= 310	6>	246	+ 49 + 15 = 310
7>	445	+ 47	+ 35	= 527	7>	445	+ 36	+ 15	= 496	7>	445	+ 36 + 15 = 496
8>	644	+ 80	+ 35	= 759	8>	644	+ 64	+ 73	= 781	8>	644	+ 67 + 15 = 726
9>	843	+ 79	+ 35	= 957	9>	843	+ 64	+ 73	= 980	9>	843	+ 67 + 15 = 925
T>	**3894**	+ **745**	+ **295**	= **4934**	T>	**3894**	+ **683**	+ **295**	= **4872**	T>	**3894**	+ **690** + **201** = **4785**

4ª fila

1>	246	+	49	+	15	=	310	1>	246	+	49	+	15	=	310	1>	246	+	61	+	15	=	322
2>	246	+	49	+	15	=	310	2>	246	+	61	+	15	=	322	2>	246	+	61	+	15	=	322
3>	246	+	61	+	15	=	322	3>	246	+	61	+	15	=	322	3>	246	+	49	+	15	=	310
4>	246	+	62	+	89	=	397	4>	246	+	49	+	48	=	343	4>	246	+	49	+	15	=	310
5>	246	+	55	+	95	=	396	5>	246	+	11	+	87	=	344	5>	246	+	55	+	15	=	316
6>	246	+	82	+	121	=	449	6>	246	+	82	+	15	=	343	6>	36	+	38	+	61	=	135
7>	246	+	37	+	288	=	571	7>	246	+	36	+	76	=	358	7>	246	+	36	+	15	=	297
8>	246	+	49	+	95	=	390	8>	246	+	48	+	76	=	370	8>	246	+	49	+	15	=	310
9>	246	+	45	+	220	=	511	9>	246	+	45	+	15	=	306	9>	246	+	46	+	35	=	328
T>	**2214**	+	**489**	+	**953**	=	**3656**	T>	**2214**	+	**442**	+	**362**	=	**3018**	T>	**2005**	+	**444**	+	**201**	=	**2650**

1>	184	+	70	+	67	=	321	1>	215	+	59	+	15	=	289	1>	223	+	19	+	34	=	276
2>	184	+	49	+	15	=	248	2>	215	+	23	+	17	=	255	2>	223	+	49	+	17	=	289
3>	184	+	49	+	15	=	248	3>	215	+	23	+	16	=	254	3>	224	+	49	+	10	=	283
4>	246	+	46	+	15	=	307	4>	246	+	55	+	15	=	216	4>	246	+	16	+	3	=	265
5>	246	+	23	+	16	=	285	5>	246	+	49	+	15	=	310	5>	246	+	49	+	15	=	310
6>	246	+	49	+	11	=	306	6>	246	+	49	+	15	=	310	6>	37	+	25	+	56	=	118
7>	246	+	20	+	37	=	286	7>	201	+	23	+	62	=	286	7>	204	+	23	+	37	=	264
8>	246	+	27	+	49	=	322	8>	201	+	26	+	7	=	234	8>	189	+	30	+	20	=	239
9>	246	+	25	+	39	=	310	9>	201	+	25	+	14	=	240	9>	205	+	29	+	20	=	254
T>	**2028**	+	**358**	+	**247**	=	**2633**	T>	**1986**	+	**332**	+	**176**	=	**2494**	T>	**1797**	+	**289**	+	**212**	=	**2298**

1>	196	+	18	+	91	=	305	1>	228	+	19	+	5	=	252	1>	246	+	18	+	27	=	291
2>	196	+	24	+	44	=	264	2>	228	+	25	+	23	=	276	2>	246	+	24	+	48	=	318
3>	196	+	23	+	44	=	263	3>	228	+	24	+	13	=	265	3>	247	+	23	+	15	=	285
4>	246	+	12	+	23	=	281	4>	246	+	16	+	8	=	270	4>	246	+	15	+	3	=	264
5>	246	+	23	+	37	=	306	5>	246	+	23	+	10	=	279	5>	246	+	23	+	32	=	301
6>	246	+	23	+	62	=	331	6>	246	+	24	+	14	=	284	6>	247	+	24	+	28	=	299
7>	246	+	49	+	15	=	310	7>	246	+	49	+	15	=	310	7>	246	+	48	+	15	=	309
8>	246	+	49	+	15	=	310	8>	246	+	48	+	15	=	309	8>	246	+	48	+	15	=	309
9>	246	+	48	+	15	=	309	9>	246	+	48	+	15	=	309	9>	247	+	48	+	15	=	310
T>	**2064**	+	**269**	+	**346**	=	**2679**	T>	**2160**	+	**276**	+	**118**	=	**2554**	T>	**2217**	+	**271**	+	**198**	=	**2686**

5ª fila

1>	246	+	12	+	21	=	279	1>	246	+	59	+	15	=	320	1>	246	+	12	+	16	=	274
2>	246	+	49	+	15	=	310	2>	246	+	65	+	87	=	398	2>	246	+	49	+	55	=	350
3>	246	+	49	+	15	=	310	3>	246	+	62	+	67	=	375	3>	246	+	12	+	62	=	320
4>	246	+	55	+	15	=	316	4>	246	+	55	+	15	=	316	4>	246	+	11	+	20	=	277
5>	246	+	62	+	67	=	375	5>	246	+	49	+	15	=	310	5>	246	+	49	+	15	=	310
6>	246	+	49	+	91	=	386	6>	246	+	49	+	15	=	310	6>	36	+	26	+	103	=	165
7>	246	+	61	+	95	=	402	7>	246	+	62	+	25	=	333	7>	246	+	56	+	67	=	369
8>	246	+	27	+	6	=	279	8>	246	+	26	+	22	=	294	8>	246	+	30	+	8	=	284
9>	246	+	25	+	7	=	278	9>	246	+	25	+	6	=	277	9>	247	+	29	+	16	=	292
T>	**2214**	+	**389**	+	**332**	=	**2935**	T>	**2214**	+	**452**	+	**267**	=	**2933**	T>	**2005**	+	**274**	+	**362**	=	**2641**

1>	184	+	97	+	28	=	309	1>	215	+	43	+	39	=	297	1>	223	+	31	+	67	=	321
2>	184	+	88	+	8	=	280	2>	215	+	56	+	40	=	311	2>	223	+	56	+	10	=	289
3>	184	+	21	+	15	=	220	3>	215	+	19	+	36	=	270	3>	224	+	11	+	11	=	246
4>	246	+	11	+	13	=	270	4>	246	+	43	+	26	=	315	4>	246	+	31	+	28	=	305
5>	246	+	21	+	15	=	282	5>	246	+	19	+	13	=	278	5>	246	+	10	+	13	=	269
6>	246	+	21	+	15	=	282	6>	246	+	19	+	10	=	275	6>	37	+	47	+	60	=	144
7>	246	+	31	+	15	=	292	7>	201	+	21	+	15	=	237	7>	204	+	12	+	15	=	231
8>	246	+	31	+	15	=	292	8>	201	+	14	+	15	=	230	8>	189	+	13	+	15	=	217
9>	246	+	29	+	15	=	290	9>	201	+	14	+	15	=	230	9>	205	+	13	+	15	=	233
T>	**2028**	+	**350**	+	**139**	=	**2517**	T>	**1986**	+	**248**	+	**209**	=	**2443**	T>	**1797**	+	**224**	+	**234**	=	**2255**

1>	196	+	29	+	15	=	240	1>	228	+	24	+	15	=	267	1>	246	+	24	+	31	=	301
2>	196	+	19	+	15	=	230	2>	228	+	36	+	23	=	287	2>	246	+	36	+	26	=	308
3>	196	+	47	+	26	=	269	3>	228	+	21	+	25	=	274	3>	247	+	21	+	47	=	315
4>	246	+	27	+	48	=	321	4>	246	+	24	+	31	=	301	4>	246	+	24	+	15	=	285
5>	246	+	43	+	26	=	315	5>	246	+	20	+	40	=	306	5>	246	+	21	+	67	=	334
6>	246	+	46	+	43	=	335	6>	246	+	20	+	25	=	291	6>	247	+	21	+	59	=	327
7>	246	+	20	+	20	=	286	7>	246	+	25	+	36	=	307	7>	246	+	25	+	37	=	308
8>	246	+	29	+	20	=	295	8>	246	+	25	+	20	=	291	8>	246	+	27	+	18	=	291
9>	246	+	36	+	14	=	296	9>	246	+	37	+	68	=	351	9>	247	+	40	+	40	=	327
T>	**2064**	+	**296**	+	**227**	=	**2587**	T>	**2160**	+	**232**	+	**283**	=	**2675**	T>	**2217**	+	**239**	+	**340**	=	**2796**

6ª fila

1> 246 + 11 + 15 = 272	1> 246 + 12 + 31 = 289	1> 246 + 11 + 16 = 273
2> 246 + 49 + 25 = 320	2> 246 + 52 + 21 = 319	2> 246 + 51 + 12 = 309
3> 246 + 48 + 25 = 319	3> 246 + 51 + 52 = 349	3> 246 + 10 + 62 = 318
4> 246 + 10 + 51 = 307	4> 246 + 11 + 20 = 277	4> 246 + 10 + 20 = 276
5> 246 + 46 + 67 = 359	5> 246 + 49 + 32 = 327	5> 246 + 48 + 32 = 326
6> 246 + 48 + 25 = 319	6> 246 + 51 + 10 = 307	6> 36 + 26 + 43 = 105
7> 246 + 49 + 15 = 310	7> 246 + 49 + 15 = 310	7> 246 + 48 + 15 = 309
8> 246 + 49 + 15 = 310	8> 246 + 48 + 15 = 309	8> 246 + 48 + 15 = 309
9> 246 + 48 + 15 = 309	9> 246 + 48 + 15 = 309	9> 247 + 48 + 15 = 310
T> **2214** + **358** + **253** = **2825**	T> **2214** + **371** + **211** = **2796**	T> **2005** + **300** + **230** = **2535**

1> 184 + 19 + 15 = 218	1> 215 + 27 + 15 = 257	1> 223 + 27 + 10 = 260
2> 184 + 27 + 15 = 226	2> 215 + 204 + 31 = 450	2> 223 + 204 + 30 = 457
3> 184 + 23 + 40 = 247	3> 215 + 23 + 44 = 282	3> 224 + 24 + 42 = 290
4> 246 + 9 + 10 = 265	4> 246 + 27 + 10 = 283	4> 246 + 27 + 15 = 288
5> 246 + 23 + 40 = 309	5> 246 + 23 + 17 = 286	5> 246 + 23 + 37 = 306
6> 246 + 22 + 33 = 301	6> 246 + 22 + 62 = 330	6> 37 + 361 + 27 = 425
7> 246 + 67 + 30 = 343	7> 201 + 100 + 28 = 329	7> 204 + 100 + 67 = 371
8> 246 + 98 + 8 = 352	8> 201 + 121 + 27 = 349	8> 189 + 124 + 39 = 352
9> 246 + 157 + 6 = 409	9> 201 + 187 + 50 = 438	9> 205 + 191 + 31 = 427
T> **2028** + **445** + **197** = **2670**	T> **1986** + **734** + **284** = **3004**	T> **1797** +**1081** + **298** = **3176**

1> 196 + 433 + 28 = 657	1> 228 + 466 + 25 = 719	1> 246 + 114 + 67 = 427
2> 196 + 444 + 22 = 662	2> 228 + 181 + 8 = 417	2> 246 + 180 + 8 = 434
3> 196 + 49 + 15 = 260	3> 228 + 50 + 46 = 324	3> 247 + 49 + 28 = 324
4> 246 + 433 + 24 = 703	4> 246 + 473 + 39 = 758	4> 246 + 120 + 28 = 394
5> 246 + 337 + 22 = 605	5> 246 + 49 + 15 = 310	5> 246 + 49 + 15 = 310
6> 246 + 49 + 46 = 341	6> 246 + 49 + 15 = 310	6> 247 + 49 + 15 = 311
7> 246 + 305 + 3 = 554	7> 246 + 306 + 48 = 600	7> 246 + 83 + 15 = 344
8> 246 + 379 + 17 = 642	8> 246 + 376 + 197 = 819	8> 246 + 138 + 15 = 399
9> 246 + 227 + 13 = 486	9> 246 + 227 + 15 = 488	9> 247 + 231 + 13 = 491
T> **2064** +**2656** + **190** = **4910**	T> **2160** +**2177** + **408** = **4745**	T> **2217** +**1013** + **204** = **3434**

7ª fila

1> 246 + 61 + 15 = 322	1> 246 + 61 + 15 = 322	1> 246 + 49 + 15 = 310
2> 246 + 61 + 15 = 322	2> 246 + 61 + 15 = 322	2> 246 + 49 + 15 = 310
3> 246 + 62 + 67 = 375	3> 246 + 49 + 15 = 310	3> 246 + 55 + 12 = 313
4> 246 + 62 + 67 = 375	4> 246 + 49 + 15 = 310	4> 246 + 55 + 12 = 313
5> 246 + 49 + 15 = 310	5> 246 + 54 + 44 = 344	5> 246 + 54 + 40 = 340
6> 246 + 49 + 15 = 210	6> 246 + 56 + 45 = 347	6> 36 + 160 + 36 = 232
7> 246 + 48 + 155 = 449	7> 246 + 48 + 10 = 304	7> 246 + 51 + 140 = 437
8> 246 + 48 + 278 = 572	8> 246 + 48 + 73 = 367	8> 246 + 49 + 12 = 307
9> 246 + 49 + 8 = 303	9> 246 + 50 + 132 = 428	9> 247 + 51 + 67 = 365
T> **2214** + **489** + **635** = **3338**	T> **2214** + **476** + **364** = **3054**	T> **2005** + **573** + **349** = **2927**

1> 184 + 239 + 57 = 480	1> 215 + 38 + 15 = 268	1> 223 + 38 + 15 = 276
2> 184 + 246 + 30 = 460	2> 215 + 45 + 38 = 298	2> 223 + 45 + 15 = 283
3> 184 + 23 + 5 = 212	3> 215 + 49 + 15 = 279	3> 224 + 49 + 15 = 288
4> 246 + 14 + 10 = 270	4> 246 + 38 + 15 = 299	4> 246 + 39 + 22 = 307
5> 246 + 49 + 15 = 310	5> 246 + 25 + 36 = 307	5> 246 + 25 + 22 = 293
6> 246 + 49 + 15 = 310	6> 246 + 24 + 33 = 303	6> 37 + 27 + 49 = 113
7> 246 + 47 + 10 = 303	7> 201 + 37 + 7 = 245	7> 204 + 63 + 7 = 274
8> 246 + 47 + 13 = 306	8> 201 + 42 + 7 = 250	8> 189 + 45 + 8 = 242
9> 246 + 49 + 8 = 303	9> 201 + 65 + 7 = 273	9> 205 + 68 + 6 = 279
T> **2028** + **763** + **163** = **2954**	T> **1986** + **363** + **173** = **2522**	T> **1797** + **399** + **159** = **2355**

1> 196 + 96 + 49 = 341	1> 246 + 46 + 6 = 280	1> 246 + 49 + 8 = 303
2> 196 + 107 + 23 = 326	2> 246 + 49 + 37 = 314	2> 246 + 49 + 8 = 303
3> 196 + 52 + 7 = 255	3> 247 + 49 + 7 = 284	3> 247 + 49 + 5 = 301
4> 246 + 47 + 64 = 357	4> 246 + 46 + 7 = 299	4> 246 + 50 + 18 = 314
5> 246 + 49 + 8 = 303	5> 246 + 50 + 31 = 327	5> 246 + 329 + 43 = 618
6> 246 + 49 + 8 = 303	6> 247 + 52 + 31 = 329	6> 247 + 332 + 13 = 592
7> 246 + 48 + 15 = 309	7> 246 + 48 + 15 = 309	7> 246 + 51 + 15 = 312
8> 246 + 48 + 15 = 309	8> 246 + 48 + 15 = 309	8> 246 + 49 + 15 = 310
9> 246 + 49 + 15 = 310	9> 247 + 50 + 15 = 311	9> 247 + 51 + 15 = 313
T> **2064** + **545** + **204** = **2813**	T> **2160** + **438** + **164** = **2762**	T> **2217** +**1009** + **140** = **3366**

8ª fila

1>	246	+	49	+	15	=	356	1>	246	+	61	+	15	=	322	1>	246	+	49	+	15	=	310
2>	246	+	49	+	15	=	372	2>	246	+	61	+	32	=	339	2>	246	+	49	+	15	=	310
3>	246	+	49	+	15	=	372	3>	246	+	49	+	15	=	310	3>	246	+	55	+	8	=	309
4>	246	+	49	+	15	=	389	4>	246	+	49	+	15	=	310	4>	246	+	55	+	8	=	309
5>	246	+	62	+	67	=	310	5>	246	+	54	+	25	=	325	5>	246	+	54	+	36	=	336
6>	246	+	49	+	91	=	310	6>	246	+	56	+	10	=	312	6>	36	+	160	+	19	=	215
7>	36	+	150	+	119	=	323	7>	246	+	48	+	7	=	301	7>	246	+	51	+	27	=	324
8>	36	+	149	+	78	=	336	8>	246	+	48	+	19	=	313	8>	246	+	49	+	8	=	303
9>	36	+	149	+	78	=	303	9>	246	+	50	+	25	=	321	9>	247	+	51	+	17	=	315
T>	**2214**	+	**489**	+	**368**	=	**3071**	T>	**2214**	+	**476**	+	**163**	=	**2853**	T>	**2005**	+	**573**	+	**153**	=	**2731**

1>	184	+	239	+	10	=	433	1>	215	+	38	+	25	=	278	1>	223	+	38	+	15	=	276
2>	184	+	246	+	32	=	462	2>	215	+	45	+	18	=	278	2>	223	+	45	+	5	=	273
3>	184	+	23	+	37	=	244	3>	215	+	49	+	22	=	286	3>	224	+	49	+	26	=	299
4>	246	+	14	+	23	=	283	4>	246	+	38	+	6	=	290	4>	246	+	39	+	13	=	298
5>	246	+	49	+	30	=	325	5>	246	+	25	+	16	=	287	5>	246	+	25	+	37	=	308
6>	246	+	49	+	8	=	303	6>	246	+	24	+	30	=	300	6>	37	+	27	+	7	=	71
7>	246	+	47	+	8	=	301	7>	201	+	37	+	6	=	244	7>	204	+	63	+	5	=	272
8>	246	+	47	+	8	=	301	8>	201	+	42	+	6	=	249	8>	189	+	45	+	5	=	239
9>	246	+	49	+	8	=	303	9>	201	+	65	+	6	=	272	9>	205	+	68	+	5	=	278
T>	**2028**	+	**763**	+	**164**	=	**2955**	T>	**1986**	+	**363**	+	**135**	=	**2484**	T>	**1797**	+	**399**	+	**118**	=	**2314**

1>	196	+	96	+	27	=	319	1>	228	+	46	+	24	=	298	1>	246	+	49	+	8	=	303
2>	196	+	107	+	43	=	346	2>	228	+	49	+	8	=	285	2>	246	+	49	+	8	=	303
3>	196	+	52	+	27	=	275	3>	228	+	49	+	6	=	283	3>	247	+	49	+	14	=	310
4>	246	+	47	+	6	=	299	4>	246	+	46	+	6	=	298	4>	246	+	50	+	6	=	302
5>	246	+	49	+	5	=	300	5>	246	+	50	+	5	=	301	5>	246	+	329	+	26	=	601
6>	246	+	49	+	5	=	300	6>	246	+	52	+	5	=	303	6>	247	+	332	+	13	=	592
7>	246	+	48	+	14	=	308	7>	246	+	48	+	11	=	305	7>	246	+	51	+	23	=	320
8>	246	+	48	+	15	=	309	8>	246	+	48	+	11	=	305	8>	246	+	49	+	11	=	306
9>	246	+	49	+	12	=	307	9>	246	+	50	+	21	=	317	9>	247	+	51	+	19	=	317
T>	**2064**	+	**545**	+	**154**	=	**2763**	T>	**2160**	+	**438**	+	**97**	=	**2695**	T>	**2217**	+	**1009**	+	**128**	=	**3354**

9ª fila

1>	246	+	61	+	52	=	359	1>	246	+	61	+	7	=	314	1>	246	+	49	+	8	=	303
2>	246	+	61	+	72	=	379	2>	246	+	61	+	31	=	338	2>	246	+	49	+	8	=	303
3>	246	+	62	+	74	=	382	3>	246	+	49	+	7	=	302	3>	246	+	55	+	15	=	316
4>	246	+	62	+	84	=	392	4>	246	+	49	+	7	=	302	4>	246	+	55	+	15	=	316
5>	246	+	49	+	8	=	303	5>	246	+	54	+	60	=	360	5>	246	+	54	+	58	=	358
6>	246	+	49	+	8	=	303	6>	246	+	56	+	21	=	323	6>	36	+	160	+	41	=	237
7>	246	+	48	+	15	=	309	7>	246	+	48	+	15	=	309	7>	246	+	51	+	15	=	312
8>	246	+	48	+	15	=	309	8>	246	+	48	+	15	=	309	8>	246	+	49	+	15	=	310
9>	246	+	49	+	15	=	310	9>	246	+	50	+	15	=	311	9>	247	+	51	+	15	=	313
T>	**2214**	+	**489**	+	**343**	=	**3046**	T>	**2214**	+	**476**	+	**178**	=	**2868**	T>	**2005**	+	**573**	+	**190**	=	**2768**

1>	184	+	239	+	18	=	441	1>	215	+	38	+	3	=	256	1>	223	+	38	+	8	=	269
2>	184	+	246	+	35	=	465	2>	215	+	45	+	4	=	264	2>	223	+	45	+	8	=	276
3>	184	+	23	+	9	=	216	3>	215	+	49	+	7	=	271	3>	224	+	49	+	8	=	281
4>	246	+	14	+	21	=	281	4>	246	+	38	+	7	=	291	4>	246	+	39	+	10	=	295
5>	246	+	49	+	8	=	303	5>	246	+	25	+	53	=	324	5>	246	+	25	+	77	=	348
6>	246	+	49	+	8	=	303	6>	246	+	24	+	14	=	284	6>	37	+	27	+	36	=	100
7>	246	+	47	+	47	=	340	7>	201	+	37	+	27	=	265	7>	204	+	63	+	74	=	341
8>	246	+	47	+	52	=	345	8>	201	+	42	+	35	=	278	8>	189	+	45	+	22	=	256
9>	246	+	49	+	34	=	329	9>	201	+	65	+	68	=	334	9>	205	+	68	+	59	=	332
T>	**2028**	+	**763**	+	**232**	=	**3023**	T>	**1986**	+	**363**	+	**218**	=	**2567**	T>	**1797**	+	**399**	+	**302**	=	**2498**

1>	196	+	96	+	20	=	312	1>	228	+	46	+	29	=	303	1>	246	+	49	+	29	=	324
2>	196	+	107	+	22	=	325	2>	228	+	49	+	40	=	317	2>	246	+	49	+	15	=	310
3>	196	+	52	+	40	=	288	3>	228	+	49	+	40	=	317	3>	247	+	49	+	42	=	338
4>	246	+	47	+	53	=	346	4>	246	+	46	+	15	=	307	4>	246	+	50	+	31	=	327
5>	246	+	49	+	53	=	348	5>	246	+	50	+	78	=	374	5>	246	+	329	+	60	=	635
6>	246	+	49	+	15	=	310	6>	246	+	52	+	46	=	344	6>	247	+	332	+	62	=	641
7>	246	+	48	+	73	=	367	7>	246	+	48	+	40	=	334	7>	246	+	51	+	113	=	410
8>	246	+	48	+	90	=	384	8>	246	+	48	+	57	=	351	8>	246	+	49	+	32	=	327
9>	246	+	49	+	48	=	343	9>	246	+	50	+	108	=	404	9>	247	+	51	+	82	=	380
T>	**2064**	+	**545**	+	**414**	=	**3023**	T>	**2160**	+	**438**	+	**453**	=	**3051**	T>	**2217**	+	**1009**	+	**466**	=	**3692**

Pel que fa a l'emplenament progressiu de l'escaquer, seguint el criteri NATURMÍN:

1ª fila

1> 246 + 49 + 15 = 310		
2> 246 + 49 + 15 = 310	2> 246 + 49 + 15 = 310	
3> 246 + 61 + 15 = 322	3> 246 + 61 + 15 = 322	3> 246 + 49 + 15 = 310
4> 246 + 62 + 87 = 395	4> 246 + 49 + 48 = 343	4> 246 + 49 + 15 = 310
5> 246 + 36 + 95 = 377	5> 246 + 24 + 40 = 310	5> 246 + 36 + 15 = 297
6> 246 + 24 + 5 = 275	6> 246 + 24 + 40 = 310	6> 246 + 36 + 15 = 297
7> 184 + 37 + 95 = 316	7> 184 + 36 + 76 = 296	7> 184 + 36 + 15 = 235
8> 169 + 65 + 95 = 329	8> 169 + 64 + 98 = 331	8> 169 + 67 + 15 = 251
9> 164 + 64 + 95 = 323	9> 164 + 64 + 98 = 326	9> 164 + 67 + 15 = 246
T> **1993** + **447** + **517** = **2957**	T> **1747** + **371** + **430** = **2548**	T> **1501** + **340** + **105** = **1946**

4> 246 + 49 + 15 = 310		
5> 246 + 19 + 13 = 278	5> 246 + 49 + 15 = 310	
6> 246 + 62 + 15 = 323	6> 246 + 49 + 15 = 310	6> 246 + 49 + 15 = 310
7> 36 + 20 + 37 = 93	7> 36 + 27 + 43 = 106	7> 36 + 147 + 46 = 229
8> 36 + 27 + 37 = 100	8> 36 + 24 + 36 = 96	8> 36 + 150 + 32 = 218
9> 36 + 20 + 17 = 73	9> 36 + 9 + 54 = 99	9> 36 + 150 + 32 = 218
T> **846** + **197** + **134** = **1177**	T> **600** + **158** + **163** = **921**	T> **354** + **496** + **125** = **975**

7> 246 + 49 + 15 = 310		
8> 246 + 49 + 15 = 310	8> 246 + 49 + 15 = 310	
9> 246 + 48 + 15 = 309	9> 246 + 48 + 15 = 309	9> 246 + 49 + 15 = 310
T> **738** + **146** + **45** = **929**	T> **492** + **97** + **30** = **619**	T> 246 + 49 + 15 = 310

2ª fila

4> 246 + 49 + 15 = 310		
5> 246 + 62 + 67 = 375	5> 246 + 49 + 15 = 310	
6> 246 + 49 + 91 = 386	6> 246 + 49 + 15 = 310	6> 246 + 49 + 15 = 310
7> 36 + 150 + 119 = 305	7> 36 + 150 + 43 = 229	7> 36 + 160 + 50 = 246
8> 36 + 149 + 78 = 263	8> 36 + 150 + 42 = 228	8> 36 + 160 + 50 = 246
9> 36 + 149 + 78 = 263	9> 36 + 149 + 50 = 235	9> 36 + 160 + 50 = 246
T> **846** + **608** + **448** = **1902**	T> **600** + **547** + **165** = **1312**	T> **354** + **529** + **165** = **1048**

1< 62 + 0 + 0 = 62	1< 15 + 0 + 0 = 15	1< 5 + 0 + 0 = 5
2< 62 + 0 + 0 = 62	2< 15 + 0 + 0 = 15	2< 5 + 0 + 0 = 5
3< 62 + 0 + 0 = 62	3< 15 + 0 + 0 = 15	3< 5 + 0 + 0 = 5
7> 60 + 49 + 15 = 124		
8> 60 + 49 + 15 = 124	8> 15 + 49 + 15 = 79	
9> 60 + 41 + 15 = 116	9> 15 + 47 + 15 = 77	9> 0 + 49 + 15 = 64
T> **366** + **139** + **45** = **550**	T> **75** + **96** + **30** = **201**	T> **15** + **49** + **15** = **79**

1> 0 + 49 + 15 = 64		
2> 0 + 56 + 15 = 71	2> 0 + 49 + 15 = 64	
3> 0 + 68 + 26 = 94	3> 0 + 49 + 28 = 77	3> 0 + 49 + 15 = 64
T> **0** + **173** + **56** = **229**	T> **0** + **98** + **43** = **141**	T> 0 + 49 + 15 = 64

3ª fila

7> 0 + 49 + 15 = 64		
8> 0 + 49 + 15 = 64	8> 0 + 49 + 15 = 64	
9> 0 + 48 + 15 = 63	9> 0 + 48 + 15 = 63	9> 0 + 49 + 15 = 64
T> **0** + **146** + **45** = **191**	T> **0** + **97** + **30** = **127**	T> 0 + 49 + 15 = 64

1> 0 + 49 + 15 = 64		
2> 0 + 56 + 15 = 71	2> 0 + 49 + 15 = 64	
3> 0 + 12 + 41 = 53	3> 0 + 49 + 9 = 58	3> 0 + 49 + 15 = 64
T> **0** + **117** + **71** = **188**	T> **0** + **98** + **24** = **122**	T> 0 + 49 + 15 = 64

4> 0 + 49 + 15 = 64		
5> 0 + 65 + 36 = 101	5> 0 + 49 + 15 = 64	
6> 0 + 64 + 15 = 79	6> 0 + 49 + 15 = 64	6> 0 + 49 + 15 = 64
T> **0** + **178** + **66** = **244**	T> **0** + **98** + **30** = **128**	T> 0 + 49 + 15 = 64

4ª fila

					1>	0	+ 49	+ 15	= 64						
2>	0	+ 49	+ 15	= 64											
3>	0	+ 61	+ 15	= 76	3>	0	+ 61	+ 15	= 76						
					4>	0	+ 49	+ 48	= 97	4>	0	+ 49	+ 15	=	64
5>	0	+ 55	+ 95	= 150						5>	0	+ 55	+ 15	=	70
6>	0	+ 82	+ 121	= 203	6>	0	+ 82	+ 15	= 97						
					7>	0	+ 36	+ 76	= 112	7>	0	+ 36	+ 15	=	51
8>	0	+ 49	+ 95	= 144						8>	0	+ 49	+ 15	=	64
9>	0	+ 45	+ 220	= 265	9>	0	+ 45	+ 15	= 60						
T>	**0**	+ **341**	+ **561**	= **902**	T>	**0**	+ **322**	+ **184**	= **506**	T>	**0**	+ **189**	+ **60**	=	**249**

3>	0	+ 49	+ 15	= 64											
5>	0	+ 23	+ 16	= 39						5>	0	+ 49	+ 15	=	64
6>	0	+ 49	+ 11	= 60	6>	0	+ 49	+ 15	= 64						
					7>	0	+ 23	+ 62	= 85	7>	0	+ 23	+ 37	=	60
8>	0	+ 27	+ 49	= 76						8>	0	+ 30	+ 20	=	50
9>	0	+ 25	+ 39	= 64	9>	0	+ 25	+ 14	= 39						
T>	**0**	+ **173**	+ **130**	= **303**	T>	**0**	+ **97**	+ **91**	= **188**	T>	**0**	+ **102**	+ **72**	=	**174**

					7<	0	+ 1	+ 0	= 1	7>	0	+ 48	+ 15	=	63
8>	0	+ 49	+ 15	= 64											
9>	0	+ 48	+ 15	= 63	9>	0	+ 48	+ 15	= 63						
T>	**0**	+ **97**	+ **30**	= **127**	T>	0	+ 49	+ 15	= 64	T>	0	+ 48	+ 15	=	63

5ª fila

3>	0	+ 48	+ 15	= 63											
5>	0	+ 61	+ 67	= 128						5>	0	+ 48	+ 15	=	63
6>	0	+ 48	+ 91	= 139	6>	0	+ 48	+ 15	= 63						
					7>	0	+ 61	+ 25	= 86	7>	0	+ 55	+ 67	=	122
8>	0	+ 26	+ 6	= 32						8>	0	+ 29	+ 8	=	37
9>	0	+ 24	+ 7	= 31	9>	0	+ 24	+ 6	= 30						
T>	**0**	+ **207**	+ **186**	= **393**	T>	**0**	+ **133**	+ **46**	= **179**	T>	**0**	+ **132**	+ **90**	=	**222**

					1<	0	+ 7	+ 0	= 7	1<	0	+ 3	+ 0	=	3
2<	0	+ 18	+ 0	= 18						2<	0	+ 1	+ 0	=	1
					4<	0	+ 7	+ 0	= 7	4<	0	+ 3	+ 0	=	3
					7<	0	+ 8	+ 0	= 8	7>	0	+ 1	+ 15	=	16
8>	0	+ 30	+ 15	= 45											
9>	0	+ 28	+ 15	= 43	9>	0	+ 8	+ 15	= 23						
T>	**0**	+ **76**	+ **30**	= **106**	T>	**0**	+ **30**	+ **15**	= **45**	T>	**0**	+ **8**	+ **15**	=	**23**

					1>	0	+ 1	+ 15	= 16						
2>	0	+ 1	+ 15	= 16											
					4>	0	+ 1	+ 31	= 32	4>	0	+ 1	+ 15	=	16
T>	0	+ 1	+ 15	= 16	T>	**0**	+ **2**	+ **46**	= **48**	T>	0	+ 1	+ 15	=	16

6ª fila

					7<	0	+ 1	+ 0	= 1	7>	0	+ 0	+ 15	=	15
8>	0	+ 1	+ 15	= 16											
9>	0	+ 0	+ 15	= 15	9>	0	+ 0	+ 15	= 15						
T>	**0**	+ **1**	+ **30**	= **31**	T>	0	+ 1	+ 15	= 16	T>	0	+ 0	+ 15	=	15

					1>	0	+ 0	+ 15	= 15						
2>	0	+ 0	+ 15	= 15											
					4>	0	+ 0	+ 10	= 10	4>	0	+ 0	+ 15	=	15
T>	0	+ 0	+ 15	= 15	T>	**0**	+ **0**	+ **25**	= **25**	T>	0	+ 0	+ 15	=	15

3>	0	+ 0	+ 15	= 15											
5<	0	+ 3	+ 0	= 3						5>	0	+ 0	+ 15	=	15
6>	0	+ 0	+ 46	= 46	6>	0	+ 0	+ 15	= 15						
T>	**0**	+ **3**	+ **61**	= **64**	T>	0	+ 0	+ 15	= 15	T>	0	+ 0	+ 15	=	15

7ª fila

		1> 0 + 0 + 15 = 15
		2> 0 + 0 + 15 = 15
	3> 0 + 0 + 15 = 15	
	4> 0 + 0 + 15 = 15	
5> 0 + 0 + 15 = 15		
6> 0 + 0 + 15 = 15		
	7> 0 + 0 + 10 = 10	
		8> 0 + 0 + 12 = 12
9> 0 + 0 + 8 = 8		
T> **0** + **0** + **38** = **38**	T> **0** + **0** + **40** = **40**	T> **0** + **0** + **42** = **42**

		2> 0 + 0 + 15 = 15
	4> 0 + 0 + 15 = 15	
6> 0 + 0 + 15 = 15		
	7> 0 + 0 + 7 = 7	
		8> 0 + 0 + 8 = 8
9> 0 + 0 + 8 = 8		
T> **0** + **0** + **23** = **23**	T> **0** + **0** + **22** = **22**	T> **0** + **0** + **23** = **23**

	7> 0 + 0 + 15 = 15	
		8> 0 + 0 + 15 = 15
9> 0 + 0 + 15 = 15		
T> 0 + 0 + 15 = 15	T> 0 + 0 + 15 = 15	T> 0 + 0 + 15 = 15

8ª fila

		2> 0 + 0 + 15 = 15
	4> 0 + 0 + 15 = 15	
6> 0 + 0 + 15 = 15		
	7> 0 + 0 + 7 = 7	
		8> 0 + 0 + 8 = 8
9> 0 + 0 + 8 = 8		
T> **0** + **0** + **23** = **23**	T> **0** + **0** + **22** = **22**	T> **0** + **0** + **23** = 23

		1< 0 + 0 + 3 = 3
	3< 0 + 0 + 5 = 5	
5< 0 + 0 + 7 = 7		
	7> 0 + 0 + 3 = 3	
		8> 0 + 0 + 0 = 0
9> 0 + 0 + 8 = 8		
T> 0 + 0 + 15 = 15	T> 0 + 0 + 8 = 8	T> 0 + 0 + 3 = 3

		1> 0 + 0 + 0 = 0
	3> 0 + 0 + 0 = 0	
5> 0 + 0 + 0 = 0		
T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0

9ª fila

	7> 0 + 0 + 0 = 0	
		8> 0 + 0 + 0 = 0
9> 0 + 0 + 0 = 0		
T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0

		1> 0 + 0 + 0 = 0
	3> 0 + 0 + 0 = 0	
5> 0 + 0 + 0 = 0		
T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0

		2> 0 + 0 + 0 = 0
	4> 0 + 0 + 0 = 0	
6> 0 + 0 + 0 = 0		
T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0	T> 0 + 0 + 0 = 0

Ens hem permès aquest patracol per tenir una base sòlida a què referir el discurs. En la pàgina 178 d'aquest capítol dèiem "una cosa és afirmar, com quan presentàvem NATURMÍN, que l'emplenament virtual amb què **RE-MEMBER** determina la viabilitat d'un valor coincideix amb aquesta solució, tant si la casella s'activa amb l'escaquer buit com si aquest està mig emplenat amb valors de NATURMÍN, i una altra pretendre que la trajectòria de l'emplenament sempre tindrà la mateixa longitud (ziga-zagues incloses), mesurada en retrocessos", i aportàvem com a contraexemple, la casella 1,1 (**1**) amb 246 + 49 + 15 = 310 retrocessos, i la 5,3 (**2**) amb 130 + 49 + 15 = 194. En les darreres set pàgines i mitja podem apreciar clarament les diferències entre els itineraris d'exploració de **RE-MEMBER** quan ens limitem a emplenar una casella (les primeres 4½) o quan les anem emplenant totes ordenadament (les 3 últimes), la sobtada irrupció d'un llistat que trenca les expectatives de continuïtat creades pels precedents, en cadascun dels dos blocs, i la interrelació entre la quantia i distribució dels retrocessos (longitud i traçat de la trajectòria) associats als diversos candidats (valors d'assignació) de cada llistat (casella).

Pel que fa al primer bloc (activació d'una única casella), la primera diferència en relació al segon (emplenament progressiu de l'escaquer) és la presència dels 9 candidats en primera instància, que sempre seran confirmats per **RE-MEMBER**: només faltaria que amb l'escaquer buit alguna casella ja tingués valors vetats! Aquí cal aclarir que aquesta primera col·lecció de 81 llistats és universal (volem dir que no respon a una SUDOKU-SOLUCIÓ predeterminada), a diferència de la segona: l'únic nexe amb NATURMÍN com a solució (NATURMÍN com a patró d'exploració és inherent a **RE-MEMBER**) és que s'hagin subratllat els retrocessos propis d'aquesta solució. Però abans de seguir convindria, per a no prodigar-nos en perífrasis, adoptar una terminologia més referida a l'estructura arbòria de les exploracions **RE-MEMBER**. Val a dir que, si cada casella s'associa a un nivell **n** en el qual tindrà un nombre no més gran de $\mathbf{9^n}$ representacions, constituïda cada una per un conjunt de fins a **9** valors compatibles, podem considerar el sudoku com un conjunt de 9 arbres en què cada arrel és un node que representa un valor 1... 9 en la casella 1,1 (en aquest cas cada arbre té una altura **n = 80**) o com un arbre únic en què l'arrel és un node extern a l'escaquer amb 9 fills, cadascun dels quals representa un valor 1... 9 en la casella 1,1 (en aquest cas l'arbre té una altura **n = 81**), però aquesta qüestió ens és indiferent, i als efectes del que segueix únicament ens interessa destacar que cada conjunt de nodes relatiu a una casella el suposarem ordenat numèricament d'esquerra a dreta. Si tot recorregut de l'arbre que arribi a l'última casella (en diguem nivell **n = 80** o **n = 81**) determina una SUDOKU-SOLUCIÓ, convindrem a prendre la Solució NATURMÍN com a eix central de referència i, en cada node valor/casella anomenarem subarbres esquerres els fills en què la casella següent adopta un valor menor que el de NATURMÍN, i subarbres drets els que l'adopten major. Si voleu, la caracterització esquerra-dreta es podria aplicar, en termes relatius, a qualsevol node, fos o no una baula de la cadena NATURMÍN, però usant aquests qualificatius en sentit estricte sí que podem afirmar, quasi tautològicament, que els subarbres esquerres mai no defineixen una solució, és a dir, sempre es queden curts, amb la fulla en una casella anterior a 9,9: si algun arribés a 9,9, havent-se separat per l'esquerra (amb un valor més baix) del germà situat a la ruta NATURMÍN, implicaria que ell és l'autèntic NATURMÍN. La recíproca (si un subarbre es queda curt, segur que és esquerre) no és certa, i n'hi ha prou a considerar la casella 7,6 (**3**) de NATURMÍN, on el candidat **5** > **3** no és validat per **RE-MEMBER**: caldria assignar **3** o **6** a la casella 9,6, cosa impossible per la presència d'aquests valors a 9,2 i 9,3.

Si, per fer-ho encara més planer, a aquests subarbres per l'esquerra els anomenem "branques curtes", diríem que els retrocessos deuen la seva existència precisament a les branques curtes: en no poder arribar a la casella 9,9 i treure la fulla **OK** (arribar-hi vol dir poder-li assignar algun valor) és quan iniciem la marxa enrera fins la primera casella a qui encara quedi algun subarbre a la dreta (i en aquest cas tornarem a avançar, avanç amb final feliç garantit si el subarbre és NATURMÍN) o bé fins a tornar a la casella 1,1 (cas de l'escaquer buit, del primer bloc de 81 llistats), fins la primera casella lliure (última, retrocedint, en el cas genèric d'escaquer arbitràriament emplenat, no reflectit en llistats) o fins la casella activada, quan no queden més candidats (cas d'emplenament ordenat de l'escaquer, del segon bloc de 81 llistats). Per tal d'interpretar correctament l'evolució dels retrocessos en els llistats, caldrà, doncs, estudiar l'influència de les branques curtes per l'esquerra en cadascun dels blocs.

Si amb l'escaquer buit activem una casella, les posicions que pot tenir en relació a una branca curta per l'esquerra són aquestes tres:

A- Ser anterior a la casella d'on arrenca la branca, i en aquest cas el nombre de retrocessos en l'itinerari d'exploració definit pel primer candidat vàlid a la casella activada inclou els de la branca. Seria el cas de l'activació de 3,2, provant-hi el valor **6**, en què l'aplicació del patró NATURMÍN haurà donat lloc a l'emplenament virtual de les caselles precedents amb **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **4** i **5**, sense retrocessos, però que després d'això n'experimentarà un fotimer. Per limitar-nos als retrocessos del primer requadre 9×3, n'hi haurà: **62** en la 1ª branca curta (rebuig del candidat **1**), seguida de **62** més en la 2ª (rebuig del candidat **2**) i de **62** més en la 3ª (rebuig del candidat **3**), totes tres retirades acabant a la casella 4,2, fins que el candidat **7** obté llum verda; **15** en la 4ª branca curta (rebuig del candidat **1**), seguida de **15** més en la 5ª (rebuig del candidat **2**) i de **15** més en la 6ª (rebuig del candidat **3**), totes tres retirades acabant a la casella 5,2, fins que el candidat **8** obté llum verda; **5** en la 7ª branca curta (rebuig del candidat **1**), seguida de **5** més en la 8ª (rebuig del candidat **2**) i de **5** més en la 9ª (rebuig del candidat **3**), totes tres retirades acabant a la casella 6,2, fins que el candidat **9** obté llum verda i NATURMÍN pot seguir sense més contratemps pel que fa al requadre. En total, **246** retrocessos.

B- Coincidir amb la casella d'on arrenca la branca, i en aquest cas el nombre de retrocessos no inclou els de la branca perquè, en estar la casella virtualment emplenada amb un valor compatible, l'exploració se salta la branca (o, si us ho estimeu més així, la curtcircuita). És el cas de 4,2 (**7**) que, descomptats els retrocessos de les branques 1ª, 2ª i 3ª, se'n queda amb $\mathbf{246 - (62 + 62 + 62) = 60}$.

C- Ser posterior a la casella d'on arrenca la branca, i en aquest cas el nombre de retrocessos tornarà a incloure els de la branca, tret que, per compartir alhora valor i fila, columna o requadre 3×3 amb algunes de les seves caselles o amb totes, una part de la branca o tota ella quedin excloses de l'itinerari i també s'excloguin els retrocessos corresponents. Com a cas singular dintre d'aquest darrer supòsit, si la casella activada no només compartís valor i fila, columna o requadre amb alguna de la branca sinó que en formés part, caldria distingir entre les sub-branques amb caselles incompatibles amb l'activada (i no comptar-ne els retrocessos) i les sub-branques compatibles o que la inclouen (comptant-ne els retrocessos, tret dels corresponents a aquesta casella). Si comencem per 5,2 (**8**) i 6,2 (**9**), que no comparteixen valor entre elles ni amb 4,2 (**7**), veurem que mantenen el nombre de retrocessos d'aquesta última, $\mathbf{246 - (62 + 62 + 62) = 60}$: 5,2 (**8**) té una posició C respecte a 4,2 (**7**) i A respecte a 6,2 (**9**), raó per la qual només s'estalvia els retrocessos de les branques 4ª, 5ª i 6ª; 6,2 (**9**) té una posició C respecte a 4,2 (**7**) i 5,2 (**8**), raó per la qual només s'estalvia els retrocessos de les branques 7ª, 8ª i 9ª. Però quan passem a les caselles 7,2 (**1**), 8,2 (**2**) i 9,2 (**3**), on per força s'hauran de produir les coincidències que caracteritzen la singularitat esmentada, es farà impossible de reflectir verbalment el que s'hi esdevé, i ha semblat més profitós dedicar gairebé les quatre últimes pàgines del present capítol (ja no ve d'aquí!) a representar pas a pas els **246** retrocessos del requadre inferior; de fet, el que hi representem pas a pas son els avanços i no els retrocessos, els quals es poden mesurar com a distància entre la casella on es produeix el blocatge (marcat amb una **X**, com sempre) i aquella on es reinicia la progressió en el fotograma següent, el seu nombre es plasma en les caselles en negreta i està explicitat a peu de figura. Si volguéssim saber quants retrocessos en el primer requadre té la casella 7,2 (**1**), podríem considerar la seva posició C respecte a les 9 branques, tornar als **246** i d'aquesta quantitat descomptar els retrocessos a partir d'emplenaments virtuals amb **1**, tret que es tractés de la mateixa casella 7,2, però acabarem abans si ens limitem a considerar aquest últim subconjunt de situacions (les hem subratllat amb petits asteriscs ******, per facilitar-vos-en el seguiment) i els retrocessos que ens duran d'una a l'altra, parant esment a no incloure 7,2 (**1**) en el còmput. Si ho fem així, la 1ª, 4ª i 7ª branques (que tenien **62**, **15** i **5** retrocessos) en quedaran excloses; la 2ª i 3ª (que en tenien **62** cadascuna) es quedaran amb **23** i **23** retrocessos; la 5ª i 6ª (que en tenien **15** cadascuna) es quedaran amb **6** i **6** retrocessos, i les 8ª i 9ª (que en tenien **5** cadascuna) es quedaran amb **2** i **2** retrocessos. En total, $\mathbf{23 + 23 + 6 + 6 + 2 + 2 = 62}$, i procedint anàlogament amb les caselles 8,2 (**2**) i 9,2 (**3**), els recomptes serien idèntics. Tanmateix a les caselles 1,3 (**7**), 2,3 (**8**), 3,3 (**9**), 7,3 (**4**), 8,3 (**5**) i 9,3 (**6**), ja fora de l'abast de les 9 branques tractades, tornem a tenir **246** retrocessos. Però a les caselles 4,3 (**1**), 5,3 (**2**) i 6,3 (**3**), que comparteixen requadre 3×3 amb la part inferior de les branques que consideràvem, els **246** retrocessos han baixat a **130** perquè, quan s'activi cadascuna d'elles i s'assagi el seu candidat NATURMÍN, les 9 branques quedaran reduïdes a 6, i encara escapçades. Si prenem com a referència el valor **1** a 4,3 (seria igual a 5,3 o 6,3), aquest valor ja no podrà aparèixer a 4,2, 5,2 o 6,2, raó per la qual la 1ª branca curta desapareix

i la 2ª i 3ª representaran **47** retrocessos cada una, en comptes dels **62** d'abans; la 4ª branca també desapareix i la 5ª la 6ª es queden amb **13** de **15** retrocessos; la 7ª també desapareix i només la 8ª la 9ª mantenen els **5** retrocessos d'abans (**47 + 47 + 13 + 13 + 5 + 5 = 130**). Tot això es pot comprovar fàcilment jugant amb les representacions del primer requadre 9×3 al final del capítol.

Quan una casella estigui en posició A o C (sense afectacions) respecte a totes les branques curtes per l'esquerra, el nombre de retrocessos serà **246 + 49 + 15 = 310**.

Si ara anem emplenant l'escaquer, seguint el patró NATURMÍN, l'estatus de una nova casella en relació a les branques curtes per l'esquerra se simplifica perquè, tot i seguir considerant tres posicions, la tercera no té la complexitat de C:

D- Ser anterior a la casella d'on arrenca la branca, i en aquest cas el nombre de retrocessos en l'itinerari d'exploració definit pel primer candidat vàlid a la casella activada i per totes les caselles precedents inclou els de la branca. Seria el cas de l'activació de 3,2, provant-hi el valor **6**, amb **246** retrocessos en el primer requadre que no justificarem perquè és idèntic al del supòsit A, tret que ara la trajectòria d'exploració NATURMÍN comença a la mateixa casella, en estar emplenades (no ja virtualment, sinó de debò) les precedents.

E- Coincidir amb la casella d'on arrenca la branca, i en aquest cas el nombre de retrocessos no inclou els de la branca perquè, en estar virtualment emplenada amb un valor compatible, l'exploració se la salta. És el cas de 4,2 (**7**), amb **60** retrocessos com en el supòsit B.

F- Ser posterior a la casella d'on arrencava la branca, i en aquest cas el nombre de retrocessos no inclou els de la branca perquè, estant emplenades amb valors compatibles la casella activada (virtualment) i les precedents (efectivament), l'exploració se la salta. I aquí, a diferència del que passava en el supòsit C, la casella 5,2 (**8**) tindrà **15** retrocessos (**246 − (62 + 62 + 62 + 15 + 15 + 15) = 15**), 6,2 (**9**) en tindrà **0** (**246 − (62 + 62 + 62 + 15 + 15 + 15 + 5 + 5 + 5) = 0**), i la resta de caselles del requadre 9×3 inferior es mantindrà en **0** retrocessos (recordeu que només considerem els d'aquest primer requadre).

Considerant el conjunt de totes les branques curtes que li surten per l'esquerra a l'itinerari NATURMÍN i que, a mesura que emplenem l'escaquer i l'itinerari es va convertint en solució, les noves caselles passen de D a E i de E a F en relació a cadascuna d'aquestes branques,

- Si de la nova casella no n'arrenca cap, el nombre de retrocessos en l'itinerari d'exploració definit pel primer candidat válid serà el mateix que el del primer candidat vàlid de la casella precedent.
- Si de la nova casella n'arrencava una o més, el nombre de retrocessos serà el de la casella precedent restant-li el nombre de retrocessos corresponent a aquesta branca o branques.

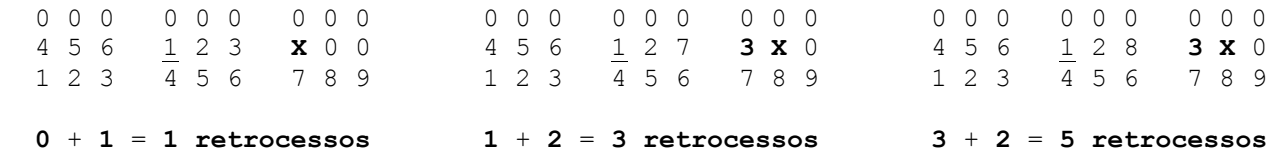
Que és una altra manera de formular aquella relació que havíem escrit subratllada. No aniria malament concretar i referir-nos als valors concrets que apareixen en les tres últimes pàgines de llistats, per tal de dissipar dubtes d'interpretació de l'enunciat. N'hi haurà prou a detallar l'inici del que en seria una correcta aplicació: els 246 + 49 + 15 = 310 retrocessos del candidat **1** a la casella 1,1 (d'ara endavant 1,1 (**1**)) es mantenen a les 11 caselles següents, 2,1 (**2**), 3,1 (**3**), 4,1 (**4**), 5,1 (**5**), 6,1 (**6**), 7,1 (**7**), 8,1 (**8**), 9,1 (**9**), 1,2 (**4**), 2,2 (**5**) i 3,2 (**6**). En les tres caselles següents, on **1**, **2** i **3** són candidats compatibles en primera instància finalment rebutjats per **RE-MEMBER**, amb 62 retrocessos per cada valor desestimat en la casella 4,2, 15 retrocessos en 5,2 i 5 retrocessos en 6,2, tots ells pertanyents al primer requadre 9×3, es dóna que: a 4,2 (**7**) els retrocessos del candidat **7** són 60 + 49 + 15 = 124, i 60 + 62 + 62 + 62 = 246, 49 + 0 + 0 + 0 = = 49 i 15 + 0 + 0 + 0 = 15; a 5,2 (**8**) els del candidat **8** són 15 + 49 + 15 = 79, i 15 + 15 + 15 + 15 = 60, 49 + 0 + 0 + 0 = 49 i 15 + 0 + 0 + 0 = 15; a 6,2 (**9**) els del candidat **9** són 0 + 49 + 15 = 64, i 0 + 5 + 5 + 5 = 15, 49 + 0 + 0 + 0 = 49 i 15 + 0 + 0 + 0 = 15. Els 0 + 49 + 15 = 64 retrocessos de l'última casella 6,2 (**9**) es mantenen 10 caselles més fins arribar a 8,4 (**9**), on la presència del candidat **7** finalment rebutjat torna a complicar les coses... Per seguir-ho més fàcilment, a cadascun dels 81 llistats que esmentàvem s'han subratllat els valors implicats.

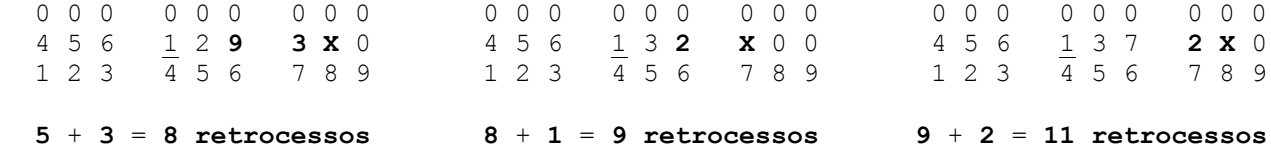
Però abans de tancar aquesta llarga digressió donant pas a la seqüència completa de representacions del requadre 9×3 inferior que havíem anunciat i que ofereix, pas a pas, els emplenaments virtuals relatius als retrocessos que s'hi produeixen, volem referir-nos a una contradicció aparent entre el primer bloc de llistats (les primeres 4½ pàgines), relatius a l'activació de caselles amb l'escaquer buit, i el que s'ha dit sobre les branques curtes que surten a l'esquerra del tronc NATURMÍN. En el fons, tot és una qüestió semàntica, referida a la qualificació "esquerra" i al poc rigor amb què hem aplicat o, més ben dit, elidit la de "curta" sense deixar

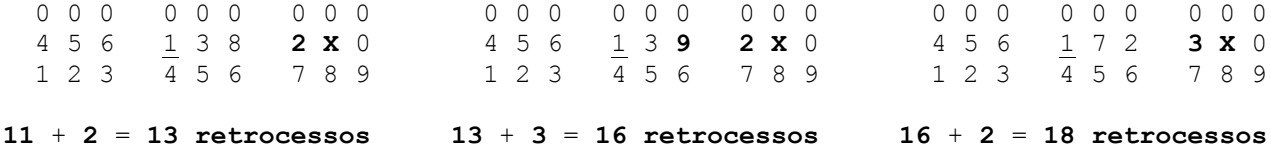
clar si era una el·lipsi retòrica per a no sobrecarregar el text o ho fèiem amb tota la idea perquè el subarbre assolia l'altura màxima. I pot confondre el fet que en el segon bloc (les 3 últimes pàgines de llistats) quan al "tronc" NATURMÍN (el subarbre de referència, com havíem dit) li surten branques per l'esquerra, que en aquell context precedeixen el valor NATURMÍN de la casella en el corresponent llistat, siguin sempre curtes (també havíem dit que la recíproca no era certa, com mostra la branca curta que surt de 7,6 amb el valor **5** > **3**, és a dir per la dreta). La diferència amb el primer bloc és que en el segon representem situacions en què, quan **RE-MEMBER** explora una casella, les anteriors ja estan ocupades: els valors inviables que ja ho eren en activar la casella precedent però s'integraven en una altre itinerari d'exploració perquè quan no es podia retrocedir més s'adoptava el candidat següent (podia ser NATURMÍN, mitjancant altres valors inviables, com quan a 4,2 es passava de **1** a **2** i de **2** a **3**, o directament, com quan es passava de **3** a **7**; o podia ser-ne un altre de complet, com quan a l'atípica 7,6 es passava de **5** a **6**), ara, en no quedar cap altre candidat, ja no tindran continuïtat. Però en el primer bloc, en treballar en un escaquer buit, ja havíem remarcat que tots els candidats en primera instància (que són els nou valors 1... 9) seran viables perquè tindran compatibilitat amb un fotimer d'itineraris complets. D'aquests, els representats pels llistats en nombre de retrocessos són els que resulten d'armonitzar el patró NATURMÍN amb el candidat sotmès a examen: si forma part de la Solució NATURMÍN la trajectòria serà el tronc de referència; si no, serà una branca llarga que li surt per la dreta (un subarbre d'altura màxima), tant si en el llistat és anterior al candidat NATURMÍN com si és posterior. Veieu, si no, les solucions corresponents als candidats **1**, **2** i **3** (en una mateixa renglera) en les caselles 4,2 (a dalt), 5,2 (al mig) i 6,2 (a baix), valors que en el segon supòsit són inviables i apareixen com a branques curtes per l'esquerra:

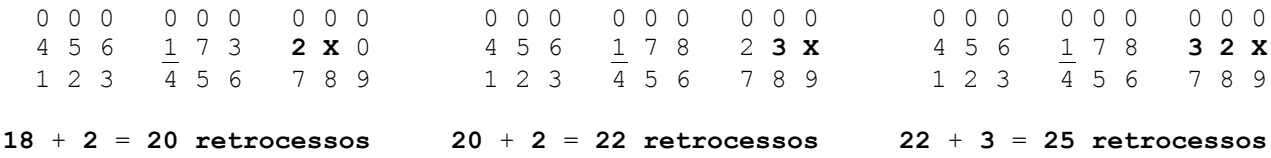
9 4 2	5 7 8	6 1 3	9 4 2	5 7 8	6 1 3	9 7 6	2 4 5	8 3 1
8 7 6	9 1 3	5 2 4	8 7 6	9 1 3	5 2 4	8 4 1	9 7 3	5 6 2
5 3 1	6 4 2	9 7 8	5 3 1	6 4 2	9 7 8	5 3 2	8 6 1	9 7 4
7 9 5	8 2 4	3 6 1	7 9 5	8 2 4	3 6 1	7 9 8	6 1 4	2 5 3
3 6 8	7 9 1	4 5 2	3 6 8	7 9 1	4 5 2	3 6 5	7 9 2	4 1 8
2 1 4	3 6 5	8 9 7	2 1 4	3 6 5	8 9 7	2 1 4	5 3 8	6 9 7
6 8 9	2 3 7	1 4 5	6 8 9	1 3 7	2 4 5	6 8 9	1 2 7	3 4 5
4 5 7	**1** 8 9	2 3 6	4 5 7	**2** 8 9	1 3 6	4 5 7	**3** 8 9	1 2 6
1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9
9 7 6	5 8 1	3 2 4	9 7 6	5 8 1	3 2 4	9 7 6	5 8 1	4 3 2
8 4 2	9 7 3	6 5 1	8 4 2	9 7 3	6 5 1	8 4 2	9 7 3	6 5 1
5 3 1	6 4 2	9 7 8	5 3 1	6 4 2	9 7 8	5 3 1	6 4 2	9 7 8
7 9 8	1 2 4	5 6 3	7 9 8	2 1 4	5 6 3	7 9 8	2 1 4	5 6 3
3 6 5	7 9 8	4 1 2	3 6 5	7 9 8	4 1 2	3 6 5	7 9 8	2 1 4
2 1 4	3 6 5	8 9 7	2 1 4	3 6 5	8 9 7	2 1 4	3 6 5	8 9 7
6 8 9	2 3 7	1 4 5	6 8 9	1 3 7	2 4 5	6 8 9	1 2 7	3 4 5
4 5 7	8 **1** 9	2 3 6	4 5 7	8 **2** 9	1 3 6	4 5 7	8 **3** 9	1 2 6
1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9
9 7 6	5 1 8	3 2 4	9 7 6	5 4 1	3 2 8	9 4 2	5 7 8	6 3 1
8 4 2	9 7 3	6 5 1	8 4 2	9 7 3	6 5 1	8 7 6	9 3 1	4 5 2
5 3 1	6 4 2	9 7 8	5 3 1	6 2 8	9 7 4	5 3 1	6 4 2	9 7 8
7 9 8	1 2 4	5 6 3	7 9 8	2 1 4	5 6 3	7 9 8	2 1 4	5 6 3
3 6 5	7 8 9	4 1 2	3 6 5	7 8 9	4 1 2	3 6 5	7 8 9	2 1 4
2 1 4	3 6 5	8 9 7	2 1 4	3 6 5	8 9 7	2 1 4	3 6 5	8 9 7
6 8 9	2 3 7	1 4 5	6 8 9	1 3 7	2 4 5	6 8 9	1 2 7	3 4 5
4 5 7	8 9 **1**	2 3 6	4 5 7	8 9 **2**	1 3 6	4 5 7	8 9 **3**	1 2 6
1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9

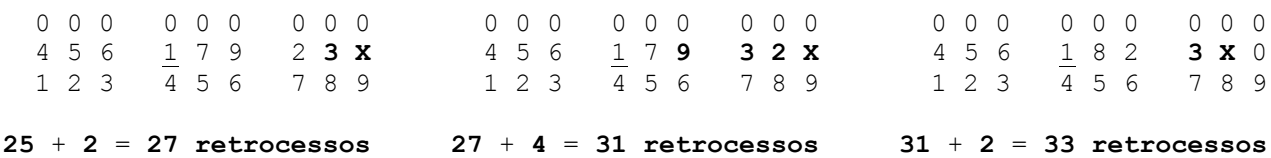
Per acabar, i tal com s'havia anunciat, els **246** retrocessos del requadre inferior:

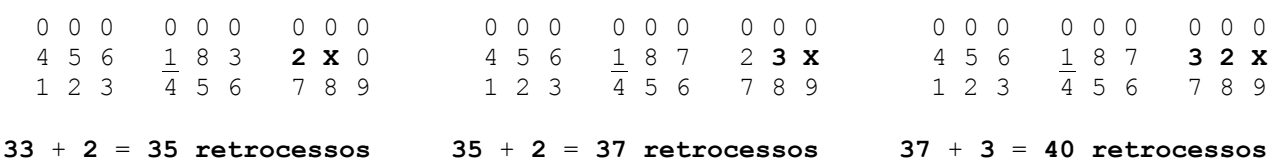

0 0 0 0 0 0 0 0 0
4 5 6 1 2 3 X 0 0
1 2 3 4 5 6 7 8 9
0 + 1 = 1 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 2 7 3 X 0
1 2 3 4 5 6 7 8 9
1 + 2 = 3 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 2 8 3 X 0
1 2 3 4 5 6 7 8 9
3 + 2 = 5 retrocessos

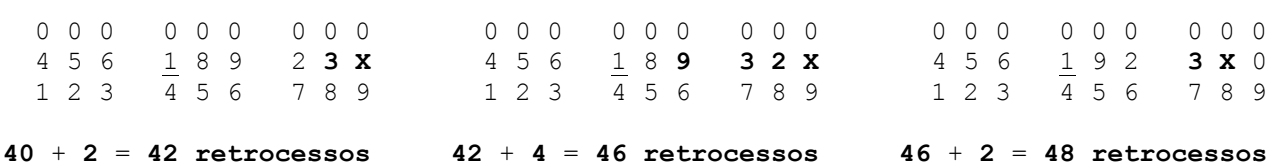

0 0 0 0 0 0 0 0 0
4 5 6 1 2 9 3 X 0
1 2 3 4 5 6 7 8 9
5 + 3 = 8 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 3 2 X 0 0
1 2 3 4 5 6 7 8 9
8 + 1 = 9 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 3 7 2 X 0
1 2 3 4 5 6 7 8 9
9 + 2 = 11 retrocessos

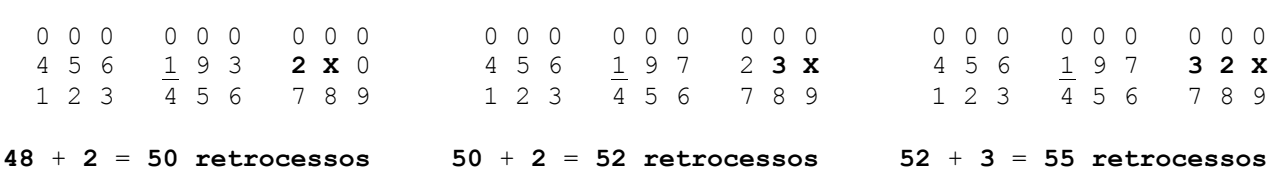

0 0 0 0 0 0 0 0 0
4 5 6 1 3 8 2 X 0
1 2 3 4 5 6 7 8 9
11 + 2 = 13 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 3 9 2 X 0
1 2 3 4 5 6 7 8 9
13 + 3 = 16 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 7 2 3 X 0
1 2 3 4 5 6 7 8 9
16 + 2 = 18 retrocessos

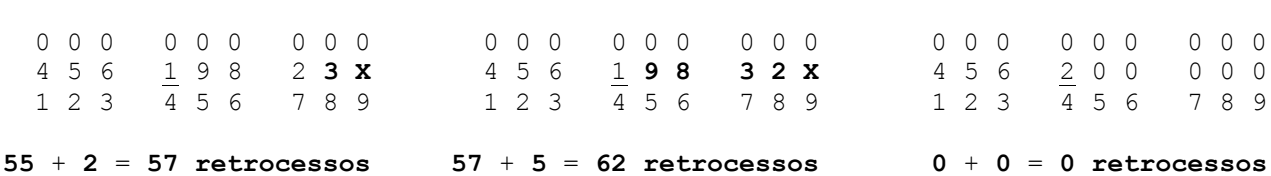

0 0 0 0 0 0 0 0 0
4 5 6 1 7 3 2 X 0
1 2 3 4 5 6 7 8 9
18 + 2 = 20 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 7 8 2 3 X
1 2 3 4 5 6 7 8 9
20 + 2 = 22 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 7 8 3 2 X
1 2 3 4 5 6 7 8 9
22 + 3 = 25 retrocessos

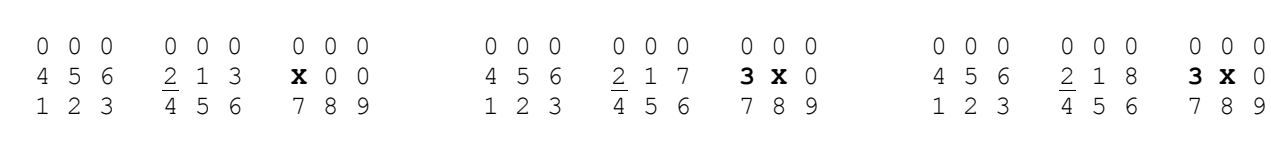

0 0 0 0 0 0 0 0 0
4 5 6 1 7 9 2 3 X
1 2 3 4 5 6 7 8 9
25 + 2 = 27 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 7 9 3 2 X
1 2 3 4 5 6 7 8 9
27 + 4 = 31 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 8 2 3 X 0
1 2 3 4 5 6 7 8 9
31 + 2 = 33 retrocessos

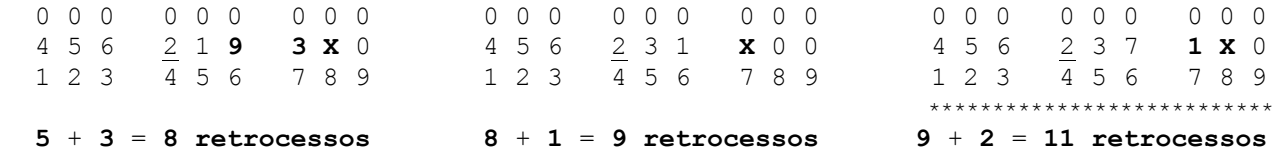

0 0 0 0 0 0 0 0 0
4 5 6 1 8 3 2 X 0
1 2 3 4 5 6 7 8 9
33 + 2 = 35 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 8 7 2 3 X
1 2 3 4 5 6 7 8 9
35 + 2 = 37 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 8 7 3 2 X
1 2 3 4 5 6 7 8 9
37 + 3 = 40 retrocessos

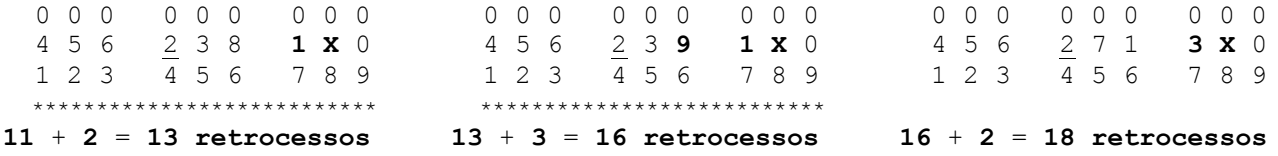

0 0 0 0 0 0 0 0 0
4 5 6 1 8 9 2 3 X
1 2 3 4 5 6 7 8 9
40 + 2 = 42 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 8 9 3 2 X
1 2 3 4 5 6 7 8 9
42 + 4 = 46 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 9 2 3 X 0
1 2 3 4 5 6 7 8 9
46 + 2 = 48 retrocessos

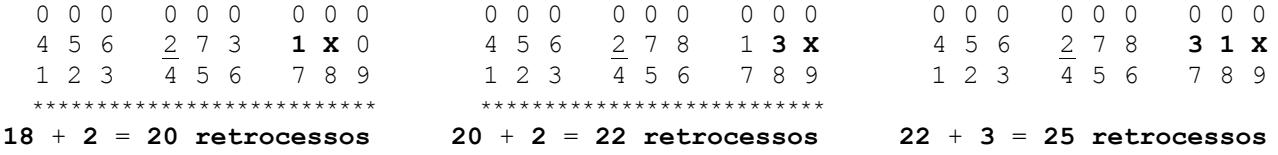

0 0 0 0 0 0 0 0 0
4 5 6 1 9 3 2 X 0
1 2 3 4 5 6 7 8 9
48 + 2 = 50 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 9 7 2 3 X
1 2 3 4 5 6 7 8 9
50 + 2 = 52 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 9 7 3 2 X
1 2 3 4 5 6 7 8 9
52 + 3 = 55 retrocessos

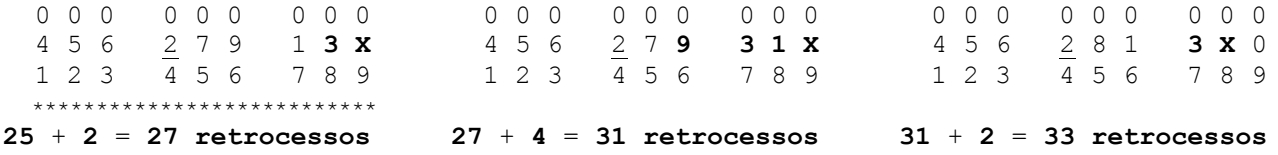

0 0 0 0 0 0 0 0 0
4 5 6 1 9 8 2 3 X
1 2 3 4 5 6 7 8 9
55 + 2 = 57 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 1 9 8 3 2 X
1 2 3 4 5 6 7 8 9
57 + 5 = 62 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 0 0 0 0 0
1 2 3 4 5 6 7 8 9
0 + 0 = 0 retrocessos

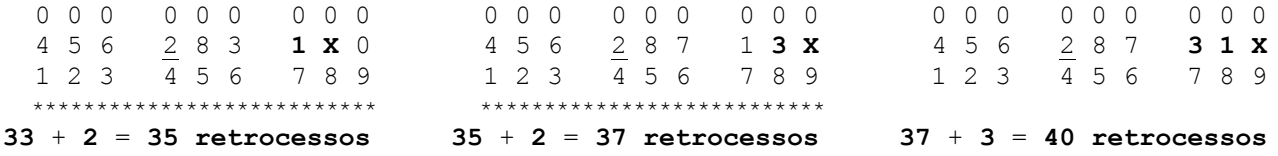

0 0 0 0 0 0 0 0 0
4 5 6 2 1 3 X 0 0
1 2 3 4 5 6 7 8 9
0 + 1 = 1 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 1 7 3 X 0
1 2 3 4 5 6 7 8 9
1 + 2 = 3 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 1 8 3 X 0
1 2 3 4 5 6 7 8 9
3 + 2 = 5 retrocessos

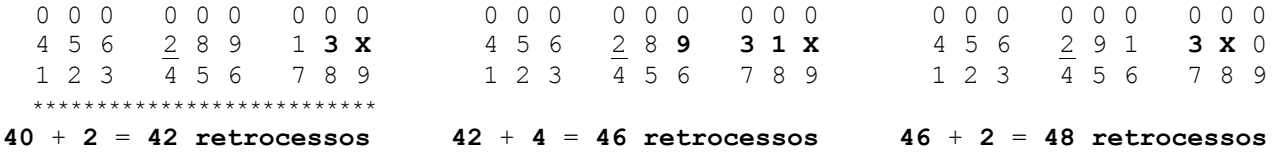

0 0 0 0 0 0 0 0 0
4 5 6 2 1 9 3 X 0
1 2 3 4 5 6 7 8 9
5 + 3 = 8 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 3 1 X 0 0
1 2 3 4 5 6 7 8 9
8 + 1 = 9 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 3 7 1 X 0
1 2 3 4 5 6 7 8 9
9 + 2 = 11 retrocessos

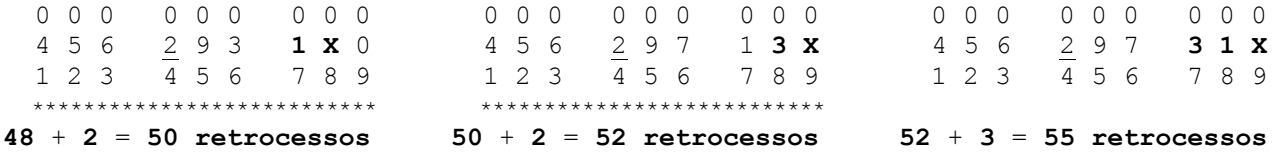

0 0 0 0 0 0 0 0 0
4 5 6 2 3 8 1 X 0
1 2 3 4 5 6 7 8 9
11 + 2 = 13 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 3 9 1 X 0
1 2 3 4 5 6 7 8 9
13 + 3 = 16 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 7 1 3 X 0
1 2 3 4 5 6 7 8 9
16 + 2 = 18 retrocessos

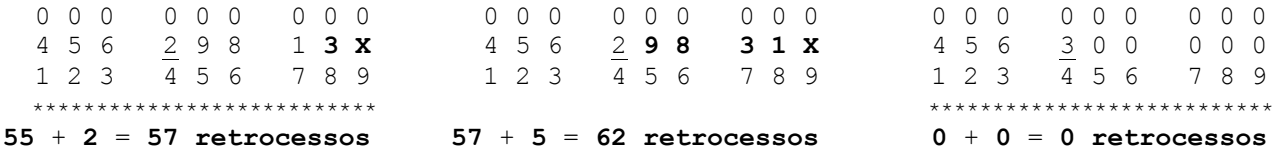

0 0 0 0 0 0 0 0 0
4 5 6 2 7 3 1 X 0
1 2 3 4 5 6 7 8 9
18 + 2 = 20 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 7 8 1 3 X
1 2 3 4 5 6 7 8 9
20 + 2 = 22 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 7 8 3 1 X
1 2 3 4 5 6 7 8 9
22 + 3 = 25 retrocessos

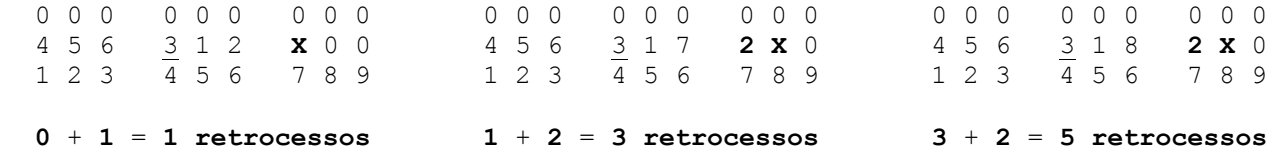

0 0 0 0 0 0 0 0 0
4 5 6 2 7 9 1 3 X
1 2 3 4 5 6 7 8 9
25 + 2 = 27 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 7 9 3 1 X
1 2 3 4 5 6 7 8 9
27 + 4 = 31 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 8 1 3 X 0
1 2 3 4 5 6 7 8 9
31 + 2 = 33 retrocessos

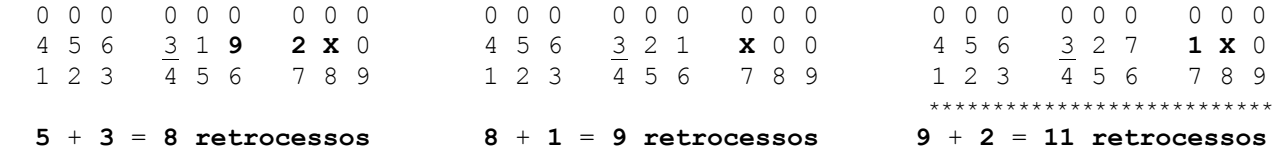

0 0 0 0 0 0 0 0 0
4 5 6 2 8 3 1 X 0
1 2 3 4 5 6 7 8 9
33 + 2 = 35 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 8 7 1 3 X
1 2 3 4 5 6 7 8 9
35 + 2 = 37 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 8 7 3 1 X
1 2 3 4 5 6 7 8 9
37 + 3 = 40 retrocessos

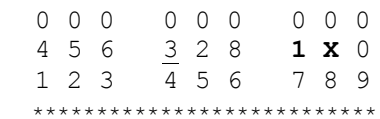

0 0 0 0 0 0 0 0 0
4 5 6 2 8 9 1 3 X
1 2 3 4 5 6 7 8 9
40 + 2 = 42 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 8 9 3 1 X
1 2 3 4 5 6 7 8 9
42 + 4 = 46 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 9 1 3 X 0
1 2 3 4 5 6 7 8 9
46 + 2 = 48 retrocessos


0 0 0 0 0 0 0 0 0
4 5 6 2 9 3 1 X 0
1 2 3 4 5 6 7 8 9
48 + 2 = 50 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 9 7 1 3 X
1 2 3 4 5 6 7 8 9
50 + 2 = 52 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 9 7 3 1 X
1 2 3 4 5 6 7 8 9
52 + 3 = 55 retrocessos


0 0 0 0 0 0 0 0 0
4 5 6 2 9 8 1 3 X
1 2 3 4 5 6 7 8 9
55 + 2 = 57 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 2 9 8 3 1 X
1 2 3 4 5 6 7 8 9
57 + 5 = 62 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 3 0 0 0 0 0
1 2 3 4 5 6 7 8 9
0 + 0 = 0 retrocessos

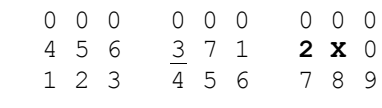

0 0 0 0 0 0 0 0 0
4 5 6 3 1 2 X 0 0
1 2 3 4 5 6 7 8 9
0 + 1 = 1 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 3 1 7 2 X 0
1 2 3 4 5 6 7 8 9
1 + 2 = 3 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 3 1 8 2 X 0
1 2 3 4 5 6 7 8 9
3 + 2 = 5 retrocessos


0 0 0 0 0 0 0 0 0
4 5 6 3 1 9 2 X 0
1 2 3 4 5 6 7 8 9
5 + 3 = 8 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 3 2 1 X 0 0
1 2 3 4 5 6 7 8 9
8 + 1 = 9 retrocessos
0 0 0 0 0 0 0 0 0
4 5 6 3 2 7 1 X 0
1 2 3 4 5 6 7 8 9
9 + 2 = 11 retrocessos

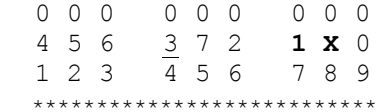


11 + 2 = 13 retrocessos

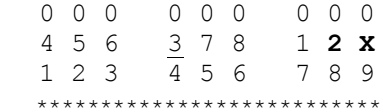
13 + 3 = 16 retrocessos



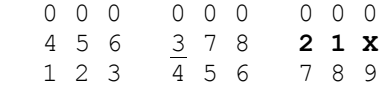
16 + 2 = 18 retrocessos



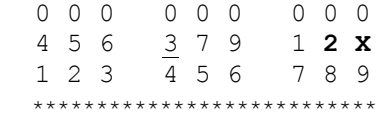
18 + 2 = 20 retrocessos



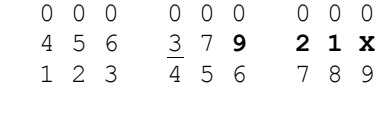
20 + 2 = 22 retrocessos



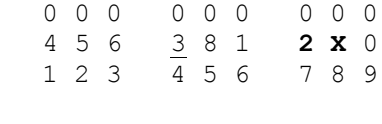
22 + 3 = 25 retrocessos



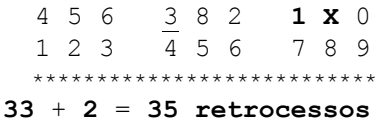
25 + 2 = 27 retrocessos



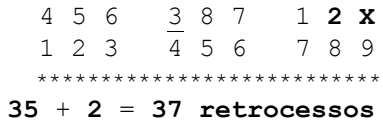
27 + 4 = 31 retrocessos



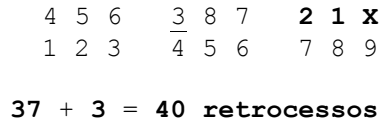
31 + 2 = 33 retrocessos



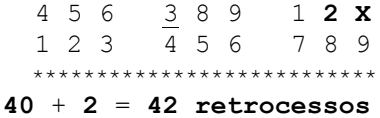
33 + 2 = 35 retrocessos



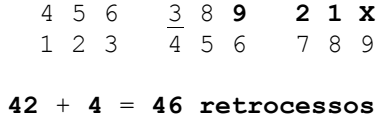
35 + 2 = 37 retrocessos



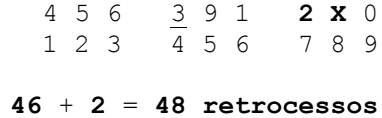
37 + 3 = 40 retrocessos



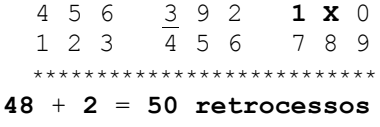
40 + 2 = 42 retrocessos



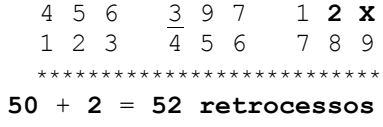
42 + 4 = 46 retrocessos



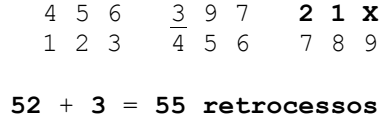
46 + 2 = 48 retrocessos



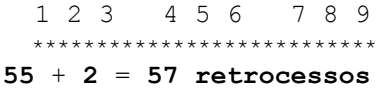
48 + 2 = 50 retrocessos



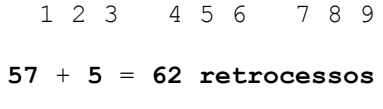
50 + 2 = 52 retrocessos



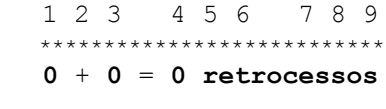
52 + 3 = 55 retrocessos



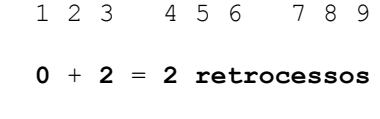
55 + 2 = 57 retrocessos



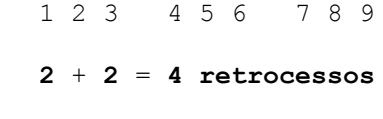
57 + 5 = 62 retrocessos



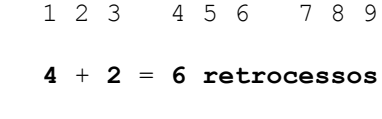
0 + 0 = 0 retrocessos



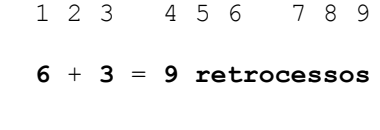
0 + 2 = 2 retrocessos



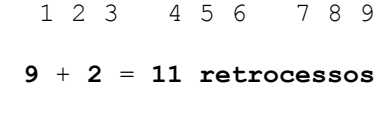
2 + 2 = 4 retrocessos



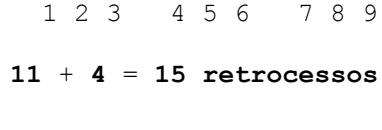
4 + 2 = 6 retrocessos



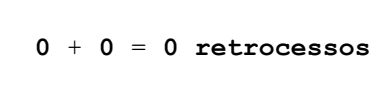
6 + 3 = 9 retrocessos



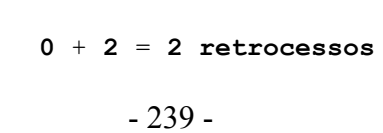
9 + 2 = 11 retrocessos



11 + 4 = 15 retrocessos



0 + 0 = 0 retrocessos



0 + 2 = 2 retrocessos

2 + 2 = 4 retrocessos

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 2 8    1 3 X
  1 2 3    4 5 6    7 8 9
  ************************
  4 + 2 = 6 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 2 8    3 1 X
  1 2 3    4 5 6    7 8 9
  ************************
  6 + 3 = 9 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 2 9    1 3 X
  1 2 3    4 5 6    7 8 9
  ************************
  9 + 2 = 11 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 2 9    3 1 X
  1 2 3    4 5 6    7 8 9
  ************************
 11 + 4 = 15 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 0    0 0 0
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 0 = 0 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 1    2 X 0
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 2 = 2 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 2    1 X 0
  1 2 3    4 5 6    7 8 9
  ************************
  2 + 2 = 4 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 8    1 2 X
  1 2 3    4 5 6    7 8 9
  ************************
  4 + 2 = 6 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 8    2 1 X
  1 2 3    4 5 6    7 8 9
  ************************
  6 + 3 = 9 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 9    1 2 X
  1 2 3    4 5 6    7 8 9
  ************************
  9 + 2 = 11 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 3 9    2 1 X
  1 2 3    4 5 6    7 8 9
  ************************
 11 + 4 = 15 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 0    0 0 0
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 0 = 0 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 1    2 3 X
  1 2 3    4 5 6    7 8 9

  0 + 2 = 2 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 1    3 2 X
  1 2 3    4 5 6    7 8 9

  2 + 3 = 5 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 2    0 0 0
  1 2 3    4 5 6    7 8 9

  0 + 0 = 0 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 2    1 3 X
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 2 = 2 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 2    3 1 X
  1 2 3    4 5 6    7 8 9
  ************************
  2 + 3 = 5 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 3    0 0 0
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 0 = 0 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 3    1 2 X
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 2 = 2 retrocessos
```

```
  0 0 0    0 0 0    0 0 0
  4 5 6    7 8 3    2 1 X
  1 2 3    4 5 6    7 8 9
  ************************
  2 + 3 = 5 retrocessos
```

```
  7 8 9    1 2 3    4 5 6
  4 5 6    7 8 9    1 2 3
  1 2 3    4 5 6    7 8 9
  ************************
  0 + 0 = 0 retrocessos
```

Etapa prèvia: crear una solució, IV (... o a Camprodon)

A l'inici del subcapítol precedent, en donar el cop de timó que comportava deixar de banda la posició de la casella actual i adoptar l'ocupació dels requadres 9×3 com a únic criteri per intercanviar-los, ens havíem oblidat que teníem en cartera la permutació de línies dins de cada requadre 9×3, també amb criteris d'ocupació. Ara, després dels successius refinaments introduïts en l'exploració **RE-MEMBER**, no hem de caure en el parany de creure que ja s'ha optimitzat tot allò optimitzable: en determinats casos, permutar les línies dels requadres 9×3 de manera que quedin ordenades de més a menys ocupació sí que reportarà guanys de temps significatius.

2 1 3 0 0 0 **7 0 4**	0 **6** 0 **4** 0 **1** 0 0 0	0 0 0 0 **2 7** **1** 0 0	0 0 0 0 **4** 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 **4** 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 **5** **4 7** 0 0 0 0	0 **4** 0 0 0 **9** 0 **5** 0	0 0 0 0 **8 5** 0 0 0	0 0 **5** **4 7** 0 0 0 0	0 **4** 0 0 0 **9** 0 **5** 0	0 0 0 0 **8 5** 0 0 0	0 0 0 0 0 **5** **4 7** 0	0 **5** 0 0 **4** 0 0 0 **9**	0 0 0 0 0 0 0 **8 5**
0 0 0 0 **4** 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	**2 1 3** 0 0 0 **7 0 4**	0 **6** 0 **4** 0 **1** 0 0 0	0 0 0 0 **2 7** **1** 0 0	**7** 0 **4** 0 0 0 **2 1 3**	0 0 0 **4** 0 **1** 0 **6** 0	**1** 0 0 0 **2 7** 0 0 0

Veiem-ho en l'exemple de l'esquerra on, d'una espera de més de 30 minuts (l'autor no ha tingut paciència d'esperar més) perquè aparegués el candidat **5** a la casella 5,4, només hem aconseguit baixar a 3 minuts i 20 segons permutant requadres 9×3 (centre), i hem hagut d'estendre l'ordenació a l'interior de cada requadre (dreta) per baixar a 20 segons. Aquí teniu tot el que ha calgut modificar perquè **MASCARA** inclogués aquesta ordenació: per evitar duplicacions hem creat la funció **PERMUT-JJ** i afegit a **F=3** el paràmetre binari **9*3**, cosa que al seu torn ens obliga a afegir un segon argument **T** a l'accés a aquesta funció des de **CAN->VAR** (com que únicament es tractava d'aquest canvi, no hem cregut necessari portar-la de nou fins aquí).

```
(defun F=3 (9*9 9*3 / 9**9)
  (setq 9**9 ())
  (foreach J JJ
    (if 9*3
        (progn
          (setq K (1- (* 3 J)))
          (repeat 3 (setq K (1+ K) 9**9 (cons (nth K 9*9) 9**9))))
        (setq 9**9 (cons (nth J 9*9) 9**9))))
  (reverse 9**9))

(defun PERMUT-JJ ()
  (setq JJ (if (and (>= (car N) (cadr N)) (>= (cadr N) (last N)))
               () ; Cas trivial identitat: '(0 1 2)
               (if (and (<= (car N) (cadr N)) (<= (cadr N) (last N)))
                   '(2 1 0)
                   (if (>= (car N) (last N))
                       (if (> (cadr N) (last N)) '(1 0 2) '(0 2 1))
                       (if (> (cadr N) (last N)) '(1 2 0) '(2 0 1)))))))

(defun MASCARA (L / INI OK Ñ JJ 0-**9*9** 0-9 0-LLL 0-L *P P* *PP*)
  (setq LL () SS (ssadd))
  (if POST
      (progn
        (setq N (N-3*3 **9*9**) Ñ (TRANSPOSAR N) N (1+2+3 N) Ñ (1+2+3 Ñ))
        (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
            (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
        (PERMUT-JJ)
        (if JJ
            (progn
              (setq 0-LLL () I -1)
              (foreach EE (F=3 **9*9** T)
```

```
              (setq 0-L () I (1+ I))
              (foreach E EE
                (setq 0-L (cons (if (atom E) E (list (car E) I)) 0-L)))
              (setq 0-LLL (cons (reverse 0-L) 0-LLL)))
            (setq 0-**9*9** (reverse 0-LLL) 0-LLL (F=3 LLL T)
                  P (list X (+ (rem Y 3)
                               (* 3 (- 3 (length (member (/ Y 3) JJ))))))))
          (setq 0-**9*9** **9*9** 0-LLL LLL))
      (setq *P () P* () J -1)
      (repeat 3
        (setq 0-9 () 0-L () N ())
        (repeat 3
          (setq J (1+ J) K 0)
          (foreach E (nth J 0-**9*9**)
            (if (or (atom E) (equal E P)) (setq K (1+ K))))
          (setq N (cons K N)
                0-9 (cons (nth J 0-**9*9**) 0-9) 0-L (cons (nth J 0-LLL) 0-L)))
        (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
        (PERMUT-JJ)
        (if JJ
            (progn
              (setq *PP* () I (- J 3))
              (if (and (> (cadr P) I) (<= (cadr P) J))
                  (setq P (list X (1+ (- J (length (member (rem (cadr P) 3)
                                                           JJ)))))))
              (foreach EE (F=3 0-9 ())
                (setq 0-9 () I (1+ I))
                (foreach E EE
                  (setq 0-9 (cons (if (atom E) E (list (car E) I)) 0-9)))
                (setq *PP* (cons (reverse 0-9) *PP*)))
              (setq *P (append *P (reverse *PP*)) P* (append P* (F=3 0-L ()))))
            (setq *P (append *P 0-9) P* (append P* 0-L))))
      (setq 0-**9*9** *P 0-LLL P*)
      (if (< (car Ñ) (caddr Ñ))
          (progn
            (setq K ())
            (foreach EE (reverse 0-**9*9**)
              (setq 0-L ())
              (foreach E EE
                (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                0-L)))
              (setq K (cons 0-L K)))
            (setq 0-**9*9** K K ())
            (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
            (setq 0-LLL K P (list (- 8 X) (cadr P)))))))
(foreach N L1A9
  (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                    "el nombre...\""))
  (if (and (member N L)
           (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
      (progn
        (if POST (TREU-RETOL))
        (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                       ((= N 2)     '(0 0.15))
                       ((= N 3)  '(0.15 0.15))
                       ((= N 4) '(-0.15 0))
                       ((= N 5)     '(0 0))
                       ((= N 6)  '(0.15 0))
                       ((= N 7) '(-0.15 -0.15))
                       ((= N 8)     '(0 -0.15))
                       (   T     '(0.15 -0.15))) (itoa N))
        (command "DESIGNA" (ssadd (entlast) SS) "")
        (setq LL (cons N LL)))
      (if POST (TREU-RETOL))))
(if POST
    (progn
      (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
      (setq P (list X Y)))))
```

Però no n'hi ha prou amb això, perquè hem descobert casos, caracteritzats pel fet de tenir algun requadre 9×3 que aquest dispositiu de reordenació no ha deixat en posició inferior (en haver-n'hi un altre de més poblat) i amb dos tercets complets que no comparteixen requadre 3×3, fila ni valor d'assignació, i en què l'ocupació de caselles d'un altre requadre 9×3, situades en un requadre 3×9 aliè als tercets esmentats, comporta una durada inacceptable pel que fa a l'exploració de candidats i, en particular, pel que fa a alguns dels candidats que acaben sent desestimats. En realitat, la descoberta d'aquesta tipologia de casos va ser casual i tampoc no no va ser immediata la detecció de què era allò específic que els identificava, i si ara anem de dret a uns exemples paradigmàtics no es pas perquè haguéssim tingut la xamba d'ensopegar-hi d'entrada, sinó per no fatigar més al lector amb anècdotes que sovint contribueixen a distreure l'atenció més que a centrar-la. De moment ens limitarem a la primera renglera i, en concret, als dos primers. Veureu que l'únic que els diferencia és que els dos requadres 9×3 inferiors estan intercanviats i, malgrat això, tots dos són "estables", en el sentit que el dispositiu reordenador no els afectarà: perquè els requadres 3×9 o 9×3 més densos se'n vagin cap avall i, dins de cadascun, les files més ocupades se situïn en posició inferior, no cal cap transposició ni permutació. ¿Perquè, doncs, l'activació de la casella 9,9, després d'emplenar les altres, no comporta cap incidència remarcable a l'exemple esquerre, però en el del mig el candidat **7** ens deixa a l'estacada? Aquest és el problema que ens mantindrà ocupats una bona estona (això aquí, perquè a l'autor el va fer anar de corcoll més de quatre mesos), però com que abans d'aquesta espera tan dilatada (la qualificarem amb l'eufemisme "superior a 5 minuts", perquè l'autor no va tenir la paciència d'esperar-se més i quantificar-la) se'n produïen altres de més curtes però que igualment superaven el llistó del que l'usuari estaria disposat a tolerar (els candidats **7**, **8** i **9** de la casella 9,7; els **7** i **8** de la 9,8, i els **1**, **3**, **4** i **5** que en 9,9 precedien **7**), i tanmateix tot indicava que responien a una problemàtica independent (a diferència de la candidatura **7**, finalment aquestes s'acceptaven), hem preferit aïllar el conflicte principal eliminant aquests efectes col·laterals. Ho hem aconseguit sense massa canvis: desplaçant 6 posicions cap avall la casella actual (**7**) i les quatre ocupades més del requadre 9×3 superior dret, el veredicte de tots els candidats, positiu o negatiu, arriba puntualment, tret del fatídic **7**: igualment inamovible, aquesta situació es representa a la dreta de la 1ª renglera.

```
0 0 0   0 0 0   0 0 7      0 0 0   0 0 0   0 0 7      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 8      0 0 0   0 0 0   0 0 8      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   2 6 9      0 0 0   0 0 0   2 6 9      0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      1 2 3   0 0 0   0 0 0      1 2 3   0 0 0   0 0 0
4 5 1   8 3 7   0 0 0      0 0 0   4 5 6   0 0 0      0 0 0   4 5 6   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 7
1 2 3   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 8
0 0 0   4 5 6   0 0 0      4 5 1   8 3 7   0 0 0      4 5 1   8 3 7   2 6 9
```

```
0 0 0   0 0 0   0 0 7      0 0 0   0 0 0   0 0 7      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 8      0 0 0   0 0 0   0 0 8      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   2 6 9      0 0 0   0 0 0   2 6 9      0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      1 2 3   0 0 0   0 0 0      1 2 3   0 0 0   0 0 0
4 5 1   8 3 7   0 0 0      0 0 0   4 5 6   0 0 0      0 0 0   4 5 6   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 7
1 2 3   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 8
0 0 0   4 5 6   0 0 0      4 5 1   8 3 7   0 0 0      4 5 1   8 3 7   2 6 9
```

Si recordeu que en les reordenacions que donen lloc a **0-**9*9**** i **0-LLL** la casella actual **P** hi compta com si ja fos ocupada, i que en les exploracions amb **RE-MEMBER** el candidat **N** hi figura en igualtat de condicions en relació als emplenaments ja consolidats, no us hauria d'estranyar el que veieu a la 2ª renglera: si **7** fos un candidat acceptat a 9,9 (esquerra i centre), podríem assegurar que, substituint **N** per **N** en qualsevol altra casella ocupada (és a dir, suposant que aquesta casella encara és buida, que l'anem a emplenar i que hi provem el candidat que havia estat

escollit a la 1ª renglera), el veredicte també seria favorable, però hem vist que, després dels altres emplenaments, **7** esdevé inviable a 9,9; així que l'única manera d'introduir-lo en aquesta casella seria deixant sense emplenar almenys una casella de les que havíem ocupat en la 1ª renglera (podria ser que calgués deixar-ne més), però en el nostre exemple es dóna la circumstància que, si deixem sense emplenar una única casella dels dos tercets abans esmentats (1ª i 2ª fila, si els situem a l'esquerra; 3ª i 4ª, si els situem al mig), ja n'hi haurà prou per posar **7** a 9,9. Més endavant entendrem per què això és així, però ara convé acabar el que dúiem entre mans: així com, quan a la 1ª renglera deixem que 9,9 sigui l'última casella per emplenar, el valor **7** no hi té cabuda, quan en la 2ª renglera emplenem 9,9 amb un **7** i deixem que la 2ª casella del segon tercet (5,1, si la situem a l'esquerra; 5,4, si la situem al mig) sigui l'última, per exemple, el candidat **5** serà refusat. Aquí es repeteix una història molt semblant a la vista en la 1ª renglera: en el cas de l'esquerra, el veredicte desfavorable del **5** (a 5,1) serà immediat; en el del centre (on és a 5,4), s'eternitza, després d'experimentar retards menors els dels candidats precedents (**1** i **2**), i en el de la dreta, el veredicte desfavorable del **5** (a 5,1) és l'única excepció en un context d'esperes perfectament raonables. Sobre els efectes col·laterals al conflicte principal en tornarem a parlar a la fi del subcapítol i, per evitar interferències, triarem exemples com els de la dreta.

Exposat el problema i delimitat el context en què es presenta, podem entrar en el terreny de les interpretacions, que més d'un lector ja haurà intuït en considerar uns exemples tan tendenciosament preparats. I tampoc no seria estrany que hagués copsat d'entrada, donant un pas més enllà, allò que l'autor no va arribar a veure fins més tard, atrapat a mig camí per un prejudici que arrossegava de lluny i que li havia inspirat les últimes implementacions efectuades: que, tal i com tenia el programa, totes les incidències que se li anessin presentant es podrien afrontar revisant la reordenació de ****9*9**** i **LLL**. Però, anant per pams, la primera cosa a remarcar del segon requadre 9×3 (a partir d'ara tot es refereix exclusivament als exemples de la dreta) era que, si consideràvem exclusivament els tres valors d'un tercet, sense tenir en compte l'ordre, dels 9 tercets de caselles de què constava el requadre només n'hi havia de tres tipus, cadascun dels quals apareixia 3 cops: dos es manifestaven en **0-**9*9**** (els dos tercets plens que anomenarem **AAA** i **BBB**) i els altres set estaven implícits en **0-LLL** (també consideràvem tercets definits les tríades de caselles buides, perquè els únics valors admissibles a cada casella eren els mateixos tres); en 4 d'aquests tercets potencials es repetien, dos a dos, els valors integrants dels tercets explícits esmentats, que anomenarem AAA i BBB, i els altres 3 tenien la mateixa composició, que coincidia amb els tres valors no presents en AAA ni a BBB, i que anomenarem CCC. És clar que això no significa que totes les llistes de **0-LLL** corresponents a cada casella buida només tinguin tres valors no nuls, gràcies a les actualitzacions que ha anat fent la funció **ACT-LLL** (de fet, només n'hi ha sis amb aquest aspecte, que són les situades a la mateixa fila i columna que les sis caselles plenes que conformen els tercets **AAA** i **BBB**), sinó que només haurien de tenir tres valors no nuls, si **ACT-LLL** fos més eficient. Copsar això és copsar que la solució al problema calia orientar-la en la direcció de més gran eficiència apuntada ... però l'autor, obcecat pel fet que en els casos de l'esquerra els veredictes dels candidats **7** i **5** venien sense demora, i convençut que tot es podia resoldre comptant les caselles buides com si fossin plenes, amb el propòsit que el dispositiu reordenador garantís (en els casos centrals, en els drets i en qualsevol altre) que un requadre 9×3 portador de configuracions **AAA-BBB** o **AAA-BBB** (que així les anomenarem a partir d'ara) acabaria en posició inferior, no hi va caure fins després d'haver muntat estratègies tan aparatoses com inútils. Però el més greu és que, no estant satisfet amb la simple constatació empírica del comportament favorable dels exemples de l'esquerra i tractant d'argumentar-lo, va estar a punt de treure'n l'entrellat però no va tornar al nus del problema, donant el pas definitiu, fins adonar-se que la via iniciada era un carreró sense sortida. ¿Quines consideracions convergien sobre aquest nus, deixant correctament plantejat el problema, en perfectes condicions d'abordatge?:

- Que les caselles buides d'un requadre 9×3 **AAA-BBB** només tenien 3 candidats.
- Que aquests candidats quedaven determinats per la configuració **AAA-BBB**.
- Que, tret de 6 caselles, les llistes **0-LLL** indicaven a la funció **RE-MEMBER** que n'hi havia més, fomentant una exploració força més llarga del que de fet calia.
- Que el punt crític de l'exploració era la casella intersecció entre una tríada de valors en columna i un tercet amb només aquests mateixos valors admissibles.
- Que l'excés de candidats de **0-LLL** impedia detectar la casella com a cul-de-sac.
- Que, alternativament, quan més avall se situés aquest punt crític, menor seria el nombre de ziga-zagues de l'exploració fins a confirmar el veredicte negatiu.

Disortadament, l'autor va prioritzar l'última consideració sobre totes les demés.

Abans de seguir, però, convindria justificar la pressumpció que, en els requadres 9×3 crítics d'aquests casos, només existeixen tres tercets (amb valors efectius o potencials) de composició diferent. Jugant amb els valors concrets dels exemples, és immediat (i aquí tant és que la 5ª casella de la figura estigui consolidada amb un **5** o que sigui la casella actual, en què explorem la viabilitat del candidat **5**):

```
0 0 0   0 0 0   0 0 0       4 5 6   1 2 3   0 0 0       4 5 6   1 2 3   7 8 9
1 2 3   7 8 9   0 0 0       1 2 3   7 8 9   4 5 6       1 2 3   7 8 9   4 5 6
7 8 9   4 5 6   0 0 0       7 8 9   4 5 6   1 2 3       7 8 9   4 5 6   1 2 3
```

A l'esquerra tenim l'evidència del que havíem avançat: que només en les 6 caselles situades en una mateixa fila i columna que alguna de les 6 ocupades **1-2-3** i **4-5-6**, les llistes de **0-LLL** tindran exactament l'aspecte **(() () () () () () 7 8 9)**, en no ser admissibles els valors **1**, **2**, **3**, **4**, **5** o **6**. En aquest sentit, les ocupacions que hem anotat en el 1r i 5è tercet, 7-8-9, són purament referencials: en qualsevol d'ambdós tercets, podrien ser també 7-9-8, 8-7-9, 8-9-7, 9-7-8 i 9-8-7, excloses les incompatibles amb ocupacions alienes al requadre representat. En el centre hi tenim la conseqüència lògica de la representació anterior: les 3 últimes caselles de la 1a fila només poden aspirar als valors 1-2-3 (amb les altres 5 permutacions) perquè els valors 7, 8, 9, **4**, **5** o **6** no poden repetir-se, tot i que les llistes de **0-LLL** incloguin alguns dels valors no escrits en negreta, perquè la humil funció **ACT-LLL** únicament té en compte les ocupacions efectives enregistrades a **0-**9*9****; les 3 últimes caselles de la 2a fila només poden aspirar als valors 4-5-6 perquè els valors **1**, **2**, **3**, 7, 8 o 9 no poden repetir-se, tot i que les llistes de **0-LLL** incloguin alguns dels valors no escrits en negreta, perquè **ACT-LLL** únicament té en compte les ocupacions **0-**9*9****; les 3 últimes caselles del 1r requadre 3×3 només poden aspirar als valors 4-5-6 perquè els valors 7, 8, 9, **1**, **2** o **3** no poden repetir-se, tot i que les llistes de **0-LLL** incloguin alguns dels valors no escrits en negreta, perquè **ACT-LLL** només té en compte les ocupacions **0-**9*9****; finalment, les 3 últimes caselles del 2n requadre 3×3 només poden aspirar als valors 1-2-3 perquè els valors **4**, **5**, **6**, 7, 8 i 9 no poden repetir-se, tot i que les llistes de **0-LLL** incloguin alguns dels valors no escrits en negreta, perquè **ACT-LLL** només té en compte les ocupacions **0-**9*9****. I a la dreta es representa el tercer i últim acte: les 3 últimes caselles de la 3a fila, que també ho són del 3r requadre 3×3, només poden aspirar als valors 7-8-9 perquè els valors 1, 2, 3, 4, 5 o 6 no poden repetir-se, tot i que les llistes de **0-LLL** incloguin alguns d'aquests valors, ja que la funció **ACT-LLL** únicament té en compte les ocupacions efectives **0-**9*9****. Si en lloc de **1-2-3**, **4-5-6** i 7-8-9 parlem de **AAA**, **BBB** i CCC, tot funciona igual. Queda clar que, en aquestes configuracions, la repetició de tercets es pot produir en la direcció de la diagonal principal, com en els casos vistos, esquemàticament,

```
BBB AAA CCC   CCC BBB AAA   AAA CCC BBB
AAA CCC BBB   BBB AAA CCC   CCC BBB AAA
CCC BBB AAA   AAA CCC BBB   BBB AAA CCC

BBB CCC AAA   CCC AAA BBB   AAA BBB CCC
CCC AAA BBB   AAA BBB CCC   BBB CCC AAA
AAA BBB CCC   BBB CCC AAA   CCC AAA BBB

BBB AAA CCC   CCC BBB AAA   AAA CCC BBB
AAA CCC BBB   BBB AAA CCC   CCC BBB AAA
CCC BBB AAA   AAA CCC BBB   BBB AAA CCC
```

o en la direcció de la diagonal secundària, esquemàticament,

```
BBB AAA CCC   AAA CCC BBB   CCC BBB AAA
CCC BBB AAA   BBB AAA CCC   AAA CCC BBB
AAA CCC BBB   CCC BBB AAA   BBB AAA CCC

AAA BBB CCC   BBB CCC AAA   CCC AAA BBB
CCC AAA BBB   AAA BBB CCC   BBB CCC AAA
BBB CCC AAA   CCC AAA BBB   AAA BBB CCC

AAA CCC BBB   CCC BBB AAA   BBB AAA CCC
BBB AAA CCC   AAA CCC BBB   CCC BBB AAA
CCC BBB AAA   BBB AAA CCC   AAA CCC BBB
```

Però les configuracions **AAA-BBB** i **AAA-B<u>B</u>B** no són les úniques que poden eternitzar l'espera del veredicte negatiu d'un candidat. Veieu, si no, els exemples arrambats a la dreta per alinear-los verticalment amb la la tercera variant dels precedents, ja que s'ha omès la primera (no conflictiva) i la segona (amb efectes col·laterals que podien distreure del conflicte principal), amb configuracions que anomenarem **AAA-AAA** i **AAA-A<u>A</u>A** perquè, a diferència de les primeres, es caracteritzen per tenir dos tercets complets (o gairebé) que no comparteixen requadre 3×3 ni fila, però sí els valors d'assignació (que un tercet sigui **7-8-9** i l'altre **9-7-8** només s'ha fet per posar ènfasi en que la composició ha de ser la mateixa, però no l'ordenació). Queda clar que en aquestes configuracions, la repetició de tercets es pot produir en la direcció de la diagonal secundària, com en aquests casos, esquemàticament,

CCC BBB AAA	BBB **AAA** CCC	**AAA** CCC BBB	0 0 0	0 0 0	0 0 0
AAA CCC BBB	CCC BBB AAA	BBB **AAA** CCC	0 0 0	0 0 0	0 0 0
BBB **AAA** CCC	**AAA** CCC BBB	CCC BBB AAA	0 0 0	0 0 0	0 0 0
CCC AAA BBB	BBB CCC **AAA**	**AAA** BBB CCC	0 0 0	0 0 0	0 0 0
BBB CCC **AAA**	**AAA** BBB CCC	CCC AAA BBB	**7 8 9**	0 0 0	0 0 0
AAA BBB CCC	CCC AAA BBB	BBB CCC **AAA**	0 0 0	**9 7 8**	0 0 0
BBB **AAA** CCC	CCC BBB AAA	AAA CCC **BBB**	0 0 0	0 0 0	0 0 **7**
AAA CCC **BBB**	BBB **AAA** CCC	CCC BBB AAA	0 0 0	0 0 0	0 0 **8**
CCC BBB AAA	AAA CCC **BBB**	BBB **AAA** CCC	**4 5 1**	**8 3 7**	**2 6 9**

o en la direcció de la principal, esquemàticament,

BBB CCC AAA	**AAA** BBB CCC	CCC **AAA** BBB	0 0 0	0 0 0	0 0 0
CCC **AAA** BBB	BBB CCC AAA	**AAA** BBB CCC	0 0 0	0 0 0	0 0 0
AAA BBB CCC	CCC **AAA** BBB	BBB CCC AAA	0 0 0	0 0 0	0 0 0
AAA CCC BBB	BBB AAA CCC	CCC BBB **AAA**	0 0 0	0 0 0	0 0 0
CCC BBB **AAA**	**AAA** CCC BBB	BBB AAA CCC	**7 8 9**	0 0 0	0 0 0
BBB AAA CCC	CCC BBB **AAA**	**AAA** CCC BBB	0 0 0	**9 <u>7</u> 8**	0 0 0
BBB CCC **AAA**	AAA BBB CCC	CCC **AAA** BBB	0 0 0	0 0 0	0 0 **7**
CCC **AAA** BBB	BBB CCC **AAA**	AAA BBB CCC	0 0 0	0 0 0	0 0 **8**
AAA BBB CCC	CCC **AAA** BBB	BBB CCC **AAA**	**4 5 1**	**8 3 7**	**2 6 9**

No us haurà passat per alt que, endut per l'entusiasme, l'autor ha omès justificar la presència de BBB i CCC com a tercets potencials que es van repetint, com havia fet amb les configuracions **AAA-BBB** i **AAA-B<u>B</u>B** (tan clar tenia que el raonament era el mateix i duria a idèntiques conclusions!) i aquest va ser el segon gran error. En resum, estava convençut que el mateix grau de determinació que es donava en les configuracions

BBB AAA CCC
CCC **BBB** AAA
AAA CCC BBB

o

BBB AAA CCC
CCC **B<u>B</u>B** AAA
AAA CCC BBB,

en què el tercet CCC no tenia cap valor 1... 9 en comú amb AAA ni amb BBB (que entre ells tampoc en tenien cap, en comú), incloent la possibilitat que la casella objecte de l'examen de candidats (anomenada indistintament casella activada o actual) fos la que faltava per completar **AAA** o **BBB**, també era el de les configuracions

BBB CCC AAA
CCC **AAA** BBB
AAA BBB CCC

o

BBB CCC AAA
CCC **A<u>A</u>A** BBB
AAA BBB CCC,

en què els dos tercets consolidats compartien els mateixos valors. Tanta confusió condemnava al fracàs el primer intent de salvar els nous obstacles.

Com que no trigareu a veure com s'ensorra aquest nou castell de cartes, tampoc no ens volem estendre en gaires explicacions: les justes per poder entendre el codi. Comentarem com l'eventualitat de disposar a la vegada de versions ordenades de les pseudomatrius habituals en posició original i transposada (per la possibilitat que

les caselles ja ocupades defineixin uns tercets organitzats segons les files, però que alguns dels valors candidats a la casella que anem a emplenar en defineixi uns altres organitzats segons les columnes i que aquesta ordenació tingui més pes que la precedent, per exemple), ha aconsellat subdividir alguns processos: la funció **TRANSPOSA-HO**, en lloc d'integrar la reestructuració de ****9*9****, **LLL** i **P**, es limita a la primera, i la reordenació de pseudomatrius tipus ****9*9**** i **LLL** se subdivideix entre les noves funcions **ORD-9*3** (permutació de les tres files d'un requadre 9×3) i **ORD-9*9** (permutació dels tres requadres 9×3 de l'escaquer 9×9). Tot això perquè, als efectes de reduir el temps d'exploració en aquests casos, seguirem la tàctica d'avançar (situar el més avall possible) els requadres 9×3 compostos per tercets que no comparteixen cap valor i que hi apareixen 3 vegades, considerant-los plens del tot: d'aquesta manera, la permutació de valors en cada tercet (que és l'únic canvi que es pot produir) no estarà condicionada al previ emplenament mecànic (en el sentit d'anar a les palpentes) del requadre o requadres 9×3 que el precedeixin.

En síntesi, obtindrem les pseudomatrius **0-9*9**, **0-LL** (versions de ****9*9**** i de **LLL** en què els requadres 9×3 estan ordenats per línies), **0-T*T** i **0-TT** (versions de les transposades en què també els requadres 9×3 estan ordenats per línies), i amb una versió modificada de la funció **N-3*3** vista en el subcapítol precedent (en què no només emmagatzarem en ***3** el nombre de caselles plenes -incloent-hi la que anem a emplenar- en els tercets dels requadres 3×3 de cada requadre 9×3, sinó els valors assignats a aquestes caselles, en ***3***) que accedeix a la nova funció **QUASI-PLE** per detectar configuracions **AAA-AAA**, **AAA-BBB**, **AAA-AAA** i **AAA-BBB** i guardar a ***P*** els valors amb aptitud per completar els tercets (de **AAA-AAA** a **AAA-AAA** i de **AAA-BBB** a **AAA-BBB**) en el cas les dues últimes i que, en sumar les ocupacions dels tercets i obtenir la dels requadres 3×3 de cada requadre 9×3 (reassignant-les a ***3** en els casos **AAA-AAA** i **AAA-BBB**, i reservant la nova variable ***A**lternativa per a **AAA-AAA** i **AAA-BBB**), totalitzant finalment l'ocupació de cada requadre 9×3 mitjançant **1+2+3**, considerarà plenes les 27 caselles dels requadres 9×3 amb tercets complets **AAA-AAA** o **AAA-BBB**. De manera que, en ordenar per requadres 9×3 **0-9*9** i **0-LL** d'una banda, i **0-T*T** i **0-TT** de l'altra, **0-**9*9**** i **0-LLL** resultaran d'un escrutini una mica més complicat que en versions precedents (en què, òbviament, l'ocupació de l'escaquer original era la mateixa del transposat i prevalia el que tenia un requadre 9×3 més ple, amb preferència per l'original si hi havia paritat en això): si l'original té la mateixa ocupació que el transposat (cosa que ara podria significar que tots dos tenen el mateix nombre de requadres 9×3 sobrevalorats i també altres possibilitats en què l'ocupació real d'uns requadres es compensa amb la sobrevaloració d'altres) seguirà dirimint la qüestió el de requadre més ocupat, amb l'original en última instància, però si l'escaquer 9×9 transposat està més ple que l'original (situació que una dràstica sobrevaloració en condicions **AAA-AAA** o **AAA-BBB** pot propiciar, en detriment d'una correlació més lligada a l'ocupació efectiva) prevaldrà el primer.

Deixant de banda els dubtes a propòsit d'una sobreponderació que pot distorsionar el mapa real d'ocupació i allunyar-nos en alguns casos de la ordenació òptima, en comptes d'acostar-nos-hi, passem a considerar la informació relativa als tercets en potència **AAA-AAA** o **AAA-BBB**, continguda a ***P*** (**QUASI-PLE** només l'haurà recollit processant **0-9*9** i **0-T*T** en el requadre 9×3 on és la casella que anem a emplenar). Serà una llista d'un o dos elements, cadascun dels quals en un primer moment en té dos: un valor binari que correspon a la variable ***C->F*** i revela si l'element següent correspon a l'escaquer original o al transposat, i una llista amb un únic valor 1... 9 (si només hem trobat configuracions **AAA-AAA**) o amb quatre (si n'hem trobat del tipus **AAA-BBB**). Paral·lelament, **N-3*3** durà el recompte ***A** de caselles plenes en els requadres 9×3, que coincidirà amb el de ***3**, tret que en el requadre no s'hi hagi detectat **AAA-AAA** ni **AAA-BBB** però sí **AAA-AAA** o **AAA-BBB**, i en aquest cas es consideraran plenes les 27 caselles (òbviament, el recompte ***A** mai no serà inferior al recompte ***3**). A diferència del que havíem fet amb les pseudomatrius **0-**9*9**** i **0-LLL** (entre l'original i la transposada, seleccionàvem la disposició que tenia una major ocupació ponderada, total o referida al requadre 9×3 més ple), en allò referent a l'ocupació ponderada que depèn de l'aptitud d'un valor candidat per completar tercets i passar de **AAA-AAA** a **AAA-AAA** i de **AAA-BBB** a **AAA-BBB**, no es farà cap tria sinó que mantindrem les dues ordenacions (cadascuna per explorar-ne la viabilitat dels candidats que figurin en l'element homòleg de ***P***), tot i que abans haurem d'aclarir una mica el panorama: anomenant **N** la llista amb l'ocupació dels 3 requadres 9×3, **NN** l'ocupació total de l'escaquer (és a dir, la suma dels 3 membres de **N**), **TN** la llista amb l'ocupació dels 3 requadres 3×9 (per ser exactes, la llista amb l'ocupació dels 3 requadres 9×3 de l'escaquer transposat, perquè amb ocupacions ponderades poden no coincidir) i **TNN** l'ocupació total de l'escaquer transposat (és a dir, la suma dels 3 membres de **TN**), tot això amb el recompte ***3**,

i anomenant **A**, **AA**, **TA** i **TAA** els homòlegs dels precedents però amb el recompte ***A**, només tindrà sentit complicar-se la vida usant **A** si **(and (> AA NN) (> AA TNN))** i només tindrà sentit complicar-se la vida usant **TA** si **(and (> TAA TNN) (> TAA NN))**. Naturalment l'esporgada s'haurà de traslladar a ***P*** i, si malgrat tot la llista encara conserva dos membres, haurem de mirar si entre els enters 1... 9 que cada un guarda n'hi ha algun de repetit (és a dir, si algun candidat a la casella que anem a emplenar determina una configuració **AAA-AAA** o **AAA-BBB** en la direcció de les files i també en la de les columnes), cas en què haurem de suprimir una d'aquestes presències: preferentment, mantindrem el valor repetit en el membre corresponent a l'escaquer més poblat. Si unes línies amunt, en descriure ***P***, havíem parlat d'una llista d'un o dos elements, cadascun dels quals en un primer moment en té dos, és perquè ara toca afegir-hi una informació que **MASCARA** necessitarà per passar-la a **RE-MEMBER**: a continuació del valor binari que revela si la informació que segueix correspon a l'escaquer original o al transposat, i de la llista amb els candidats 1... 9 que han de passar l'examen d'aquesta última funció explorant l'escaquer 9×9 reordenat d'una manera determinada, ara afegirem una tercera llista al seu torn composta per tres membres i que reflecteix aquesta reordenació, **0-9*9** o **0-T*T** amb els seus requadres 9×3 permutats al dictat de **A** o **TA**, respectivament, **0-LL** o **0-TT** tractats de la mateixa forma, i la posició de la casella que anem a emplenar en un o altre sistema. I ja sense més preàmbuls, el codi que materialitza aquest procés:

```
(defun REP-NO (L1 L2 JUNT / L)
  (foreach E L2 (if (not (member E L1)) (setq L (cons E L))))
  (if JUNT (append L1 L) (list L1 L)))

(defun QUASI-PLE (/ QPLE F-E EE FF *E *E* *F *F* E-*P*)
  (foreach EF (list '(0 1) '(0 2) '(1 2))
    (setq F-E (- (cadr EF) (car EF)) I (- (car EF) 3))
    (repeat 3
      (setq I (+ I 3) J (- (cadr EF) 3))
      (if (and (not QPLE) (= (setq *E* (nth I *3*) *E (nth I *3)) 3))
          (repeat 3 (setq J (+ J 3))
            (if (and (not QPLE) (/= J (+ I F-E)) (= (setq *F (nth J *3)) 3))
                (progn
                  (setq *F* (nth J *3*) QPLE T *A ()
                        E-*P* (member P (append *E* *F*)))
                  (if (and E-*P* (member P *F*))
                      (setq EE *F* FF *E*)
                      (setq EE *E* FF *F*))
                  (foreach E EE (if (and (atom E) (member E FF)) (setq QPLE ())))
                  (if (and QPLE E-*P*)
                      (progn
                        (foreach N L1A9
                          (if (not (member N (append EE FF)))
                              (setq *A (cons N *A))))
                        (setq E-*P* *A)))
                  (if (not QPLE)
                      (progn
                        (setq QPLE T)
                        (foreach E EE
                          (if (and (atom E) (not (member E FF)))
                              (setq QPLE ())))
                        (if (and QPLE E-*P*)
                            (foreach F FF
                              (if (not (member F EE)) (setq E-*P* (list F)))))))
                  (if QPLE
                      (if E-*P*
                          (progn
                            (setq QPLE ())
                            (if (setq *A (assoc *C->F* *P*))
                                (setq *P* (subst (list *C->F*
                                                       (REP-NO (cadr *A) E-*P* T))
                                                 *A *P*))
                                (setq E-*P* (list *C->F* E-*P*)
                                      *P* (cons E-*P* *P*))))
                          (if *P*
                              (progn
                                (setq *A *P* *P* ())
```

```
                                    (foreach E *A
                                      (if (/= (car E) *C->F*)
                                          (setq *P* (cons E *P*))))))))))))))
  (list QPLE E-*P*))

(defun N-3*3 (9*9 P *C->F* / 3*3 A*A E-*3 *3 *A E-*3* *3* I)
  (setq K -1)
  (repeat 3 ; 3 requadres 9*3
    (setq *3 () *3* ())
    (repeat 3 ; 1 requadre 9*3 = 3 files
      (setq J -1 K (1+ K))
      (repeat 3 ; 1 fila = 3 tercets
        (setq E-*3 0 E-*3* ())
        (repeat 3 ; 1 tercet = 3 caselles
          (setq J (1+ J) I (nth J (nth K 9*9)))
          (if (or (atom I) (equal I P))
              (setq E-*3 (1+ E-*3) E-*3* (cons I E-*3*))))
        (setq *3 (cons E-*3 *3) *3* (cons (reverse E-*3*) *3*))))
    (setq *3* (reverse *3*) *3 (reverse *3) E-*3 (QUASI-PLE)
          I '(9 9 9)
          J (list (+ (nth 2 *3) (nth 5 *3) (nth 8 *3))
                  (+ (nth 1 *3) (nth 4 *3) (nth 7 *3))
                  (+ (nth 0 *3) (nth 3 *3) (nth 6 *3))))
    (if (car E-*3)
        (setq *3 I *A I)
        (setq *3 J *A (if (cadr E-*3) I J)))
    (setq 3*3 (cons *3 3*3) A*A (cons *A A*A)))
  (list (reverse 3*3) (reverse A*A)))

(defun TOT (L) (if L (+ (car L) (cadr L) (last L)) 0))

(defun TRANSPOSA-HO (9*9 / 9**9)
  (foreach EE (TRANSPOSAR 9*9)
    (setq 0-L ())
    (foreach E EE (setq 0-L (cons (if (atom E) E (reverse E)) 0-L)))
    (setq 9**9 (cons (reverse 0-L) 9**9)))
  (reverse 9**9))

(defun ORD-9*3 (9*9 L*L P)
  (setq *P () P* () J -1)
  (repeat 3
    (setq 0-9 () 0-L () N ())
    (repeat 3
      (setq J (1+ J) K 0)
      (foreach E (nth J 9*9) (if (or (atom E) (equal E P)) (setq K (1+ K))))
      (setq N (cons K N) 0-9 (cons (nth J 9*9) 0-9) 0-L (cons (nth J L*L) 0-L)))
    (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
    (PERMUT-JJ)
    (if JJ
        (progn
          (setq *PP* () I (- J 3))
          (if (and (> (cadr P) I) (<= (cadr P) J))
              (setq P (list (car P)
                            (1+ (- J (length (member (rem (cadr P) 3) JJ)))))))
          (foreach EE (F=3 0-9 ())
            (setq 0-9 () I (1+ I))
            (foreach E EE
              (setq 0-9 (cons (if (atom E) E (list (car E) I)) 0-9)))
            (setq *PP* (cons (reverse 0-9) *PP*)))
          (setq *P (append *P (reverse *PP*)) P* (append P* (F=3 0-L ()))))
        (setq *P (append *P 0-9) P* (append P* 0-L))))
  (list *P P* P))

(defun ORD-9*9 (9*9 L*L P N)
  (PERMUT-JJ)
  (if JJ
      (progn
        (setq *PP* () I -1)
```

```
        (foreach EE (F=3 9*9 T)
          (setq 0-L () I (1+ I))
          (foreach E EE (setq 0-L (cons (if (atom E) E (list (car E) I)) 0-L)))
          (setq *PP* (cons (reverse 0-L) *PP*)))
        (setq *P (reverse *PP*) P* (F=3 L*L T)
              P (list (car P)
                      (+ (rem (cadr P) 3)
                         (* 3 (- 3 (length (member (/ (cadr P) 3) JJ))))))))
      (setq *P 9*9 P* L*L))
  (list *P P* P))

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N (if E (last (last E)) P) R-LLL ())
                INI (EXPLORA-3*9) R-9*9 (car 2R) R-LLL (cadr 2R)))
  (if (not (or INI OK))
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9 ; (reverse L1A9) per al patrò NATURMAX
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (foreach F (cadr 2R)
                      (if (and (not OK) (member '(() () () () () () () () ()) F))
                          (setq OK T)))
                    (if (not OK) (setq OK (EXPLORA-3*9)))
                    (if OK (setq OK ()) (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)

(defun MASCARA (L / INI OK JJ **T*T** 0-**9*9** 0-9 0-LLL 0-L 0-9*9 0-LL
                    0-T*T 0-TT TN TQ A TA NN TNN AA TAA *P* *P P* *PP*)
  (setq LL () SS (ssadd))
  (if POST
      (progn
        (setq Q (ORD-9*3 **9*9** LLL P)
              0-9*9 (car Q)  0-LL (cadr Q)  Q  (last Q)
              TQ (ORD-9*3 (TRANSPOSA-HO **9*9**) (TRANSPOSAR LLL) (reverse P))
              0-T*T (car TQ) 0-TT (cadr TQ) TQ (last TQ)
              A  (N-3*3 0-9*9  Q ()) N (car A)  A  (cadr A)
              A (if (equal A N) () A)
              TA (N-3*3 0-T*T TQ T) TN (car TA) TA (cadr TA)
              TA (if (equal TA TN) () TA)
              N (1+2+3 N) TN (1+2+3 TN)
              A (1+2+3 A) TA (1+2+3 TA)
              NN (TOT N) TNN (TOT TN) AA (TOT A) TAA (TOT TA)
              C->F (if (= NN TNN)
                       (if (< (eval (cons 'max N)) (eval (cons 'max TN))) T ())
                       (if (> TNN NN) T ()))
              P (if C->F (ORD-9*9 0-T*T 0-TT TQ TN) (ORD-9*9 0-9*9 0-LL Q N))
              0-**9*9** (car P) 0-LLL (cadr P) P (last P))
;              NN (TOT N) TNN (TOT TN) AA (TOT A) TAA (TOT TA))
;         (if (= NN TNN)
;             (if (< (eval (cons 'max N)) (eval (cons 'max TN)))
;                 (setq C->F T  P (ORD-9*9 0-T*T 0-TT TQ TN))
;                 (setq C->F () P (ORD-9*9 0-9*9 0-LL Q  N)))
;             (if (> TNN NN)
;                 (setq C->F T  P (ORD-9*9 0-T*T 0-TT TQ TN))
;                 (setq C->F () P (ORD-9*9 0-9*9 0-LL Q  N))))
;         (setq 0-**9*9** (car P) 0-LLL (cadr P) P (last P))
        (if (< (car TN) (caddr TN))
            (progn
              (setq K ())
              (foreach EE (reverse 0-**9*9**)
                (setq 0-L ())
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                  0-L)))
```

```
              (setq K (cons 0-L K)))
            (setq 0-**9*9** K K ())
            (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
            (setq 0-LLL K P (list (- 8 (car P)) (cadr P)))))
        (if *P*
            (progn
              (setq *P* (if (= (length *P*) 2) (reverse *P*) *P*)
                    *P (cadr (assoc () *P*)) P* (cadr (assoc T *P*)))
              (if (or (= AA NN) (<= AA TNN))
                  (setq  A () *P* (if P* (list (list T P*))) *P ()))
              (if (or (= TAA TNN) (<= TAA NN))
                  (setq TA () *P* (if *P (list (list () *P))) P* ()))
              (if (and A TA)
                  (progn
                    (setq P* (if (>= AA TAA) (REP-NO *P P* ()) (REP-NO P* *P ()))
                          *P (car P*) P* (cadr P*))
                    (if (not *P)
                        (setq A () *P* (list (list T P*)))
                        (if (not P*) (setq TA () *P* (list (list () *P)))))))
              (setq OK (append *P P*))
              (if A  (setq A  (ORD-9*9 0-9*9 0-LL Q A)
                           *P* (list (append (car *P*)
                                             (list (list (car A) (cadr A)
                                                         (last A))))
                                     (cadr *P*))))
              (if TA (setq A (ORD-9*9 0-T*T 0-TT TQ TA)
                           *P* (list (car *P*)
                                     (append (cadr *P*)
                                             (list (list (car A) (cadr A)
                                                         (last A)))))))
              (setq *PP* OK)))))
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ())
                 (if (member N *PP*)
                     (progn
                       (foreach E *P*
                         (if (member N (cadr E))
                             (RE-MEMBER (car (last E)) (cadr (last E)))))
                       OK)
                     (RE-MEMBER 0-**9*9** 0-LLL))))
        (progn
          (if POST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         ((= N 2)     '(0 0.15))
                         ((= N 3)  '(0.15 0.15))
                         ((= N 4) '(-0.15 0))
                         ((= N 5)     '(0 0))
                         ((= N 6)  '(0.15 0))
                         ((= N 7) '(-0.15 -0.15))
                         ((= N 8)     '(0 -0.15))
                         (   T     '(0.15 -0.15)))
                   (itoa (if MASC (nth (1- N) MASC) N)))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if (and POST (not 1-2)) (TREU-RETOL))))
  (if POST
      (progn
        (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
        (if C->F (setq C->F ()))
        (setq P (list X Y)))))
```

Fent gala d'aquest atribut tan específicament humà d'ensopegar dos cops (i tres, i quatre...) a la mateixa pedra, no va ser fins després d'haver esmerçat força dies en la consecució de les modificacions que acabem de presentar, que vam comprovar dues coses:

1) Que en el cas
BBB AAA CCC
CCC **BBB** AAA
AAA CCC BBB, l'activació d'una casella ubicada en les condicions ja esmentades (en un altre requadre 9×3 i en un requadre 3×9 aliè als dos tercets explícits) només era conflictiva si en la mateixa columna, i en requadres 9×3 aliens als tercets (en el mateix que la casella activada o en el tercer), figuraven dues caselles emplenades amb valors d'un dels dos tercets explícits (AAA o BBB) o de l'implícit (CCC). A més, orientada l'atenció cap al valor de CCC absent a la columna, es va comprovar que allò que s'eternitzava corresponia a l'exploració d'aquest valor.
2) Que el cas alternatiu
BBB CCC AAA
CCC **AAA** BBB
AAA BBB CCC, aquestes mateixes condicions no donaven lloc a cap conflicte.

¿Per què, contra tot pronòstic, **MASCARA** trigava tant en un cas a descartar aquest tercer valor, mentre que en l'altra el despatxava de seguida (o, almenys, en un temps raonable)? No va ser difícil de trobar-ne la causa: en aquests sis exemples, **AAA-BBB** els dos primers, **AAA-AAA** el tercer, **AAA-BBB** el quart i el cinquè (que es corresponen amb els situats al damunt, tret que en els superiors la casella actual és fora dels dos tercets no alineats, mentre que en els inferiors n'hi forma part) i **AAA-AAA** el sisè (que manté amb el tercer la mateixa relació que els precedents amb el primer i segon) el misteri que justifica que el veredicte de compatibilitat (que en tots sis casos és negatiu) s'emeti de seguida en els dos de la dreta però en els quatre de l'esquerra trigui més del que seria raonable esperar i haguem de recórrer a les últimes modificacions, és que **R-LLL** infringeix les **normes 2** i **3** en els dos de la dreta però les compleix escrupolosament en els quatre de l'esquerra. Com que en el subcapítol precedent havíem incorporat en **RE-MEMBER** uns accessos a **EXPLORA-3*9**, després d'introduir en aquesta segona funció la sentència condicional **(if (and INI (not FORA)) ... EXPL-SUBCONJ ()))**, la vulneració de les **normes 2** i **3** en els dos casos de la dreta ha estat detectada i s'ha interromput l'exploració, però en els quatre de l'esquerra no trobarà cap dispositiu que li estalvii ni una sola estació del viacrucis: permutant els dos primers requadres 9×3, aconseguirem que l'exploració no vagi més enllà de la casella 9,3, però mantenint les posicions que veieu arribarà fins a la casella 9,6 i, a partir d'aquí, la marxa enrera serà relativament breu pel que fa al segon requadre 9×3 (estant cantats els tres valors que han d'omplir cada tercet buit, només resta assajar-ne les permutacions) però en el primer ens hi florirem.

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 3   0 0 0   0 0 0      1 2 3   0 0 0   0 0 0      7 8 9   0 0 0   0 0 0
0 0 0   4 5 6   0 0 0      0 0 0   4 5 6   0 0 0      0 0 0   9 7 8   0 0 0

0 0 0   0 0 0   0 0 7      0 0 0   0 0 0   0 5 0      0 0 0   0 0 0   0 0 7
0 0 0   0 0 0   0 0 8      0 0 0   0 0 0   0 4 0      0 0 0   0 0 0   0 0 8
4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9



0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 3   0 0 0   0 0 0      1 2 3   0 0 0   0 0 0      7 8 9   0 0 0   0 0 0
0 0 0   4 5 6   0 0 0      0 0 0   4 5 6   0 0 0      0 0 0   9 7 8   0 0 0

0 0 0   0 0 0   0 0 7      0 0 0   0 0 0   0 5 0      0 0 0   0 0 0   0 0 7
0 0 0   0 0 0   0 0 8      0 0 0   0 0 0   0 4 0      0 0 0   0 0 0   0 0 8
4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9
```

Per verificar si es respecten o no les **normes 2** i **3**, n'hi haurà prou a estudiar el requadre 3×3 central dret de les pseudomatrius **R-LLL** corresponents a l'inici de

l'exploració de 9,3 amb **7** o de 8,3 amb **5** en els tres primers casos, ja que en els altres tres els resultas són idèntics. L'observança en els dos primers és total,

(**1** 0 **3 4 5 6 7 8 9**) (**1 2 3 4 5** 0 **7 8 9**) (**1 2 3 4 5 6** 0 0 0)

(0 0 0 **4 5 6 7 8 9**) (0 0 0 **4 5** 0 **7 8 9**) (0 0 0 **4 5 6** 0 0 0)

(**1** 0 **3** 0 0 0 **7 8 9**) (**1 2 3** 0 0 0 **7 8 9**) (**1 2 3** 0 0 0 0 0 0)

(**1** 0 **3 4 5 6 7 8 9**) (**1 2 3** 0 0 0 **7 8 9**) (**1 2 3 4 5 6 7 8** 0)

(0 0 0 **4 5 6 7 8 9**) (0 0 0 0 0 0 **7 8 9**) (0 0 0 **4 5 6 7 8** 0)

(**1** 0 **3** 0 0 0 **7 8 9**) (**1 2 3** 0 0 0 **7 8 9**) (**1 2 3** 0 0 0 **7 8** 0)

però en el tercer no: hi ha 7 llistes on només apareixen els valors **1, 2, 3, 4, 5** i **6** (6 elements), i els valors **7, 8** i **9** (3 elements) només apareixen a 2 llistes.

(**1** 0 **3 4 5 6 7 8 9**) (**1 2 3 4 5** 0 **7 8 9**) (**1 2 3 4 5 6** 0 0 0)

(**1** 0 **3 4 5 6** 0 0 0) (**1 2 3 4 5** 0 0 0 0) (**1 2 3 4 5 6** 0 0 0)

(**1** 0 **3 4 5 6** 0 0 0) (**1 2 3 4 5** 0 0 0 0) (**1 2 3 4 5 6** 0 0 0)

Algú podria argumentar que, pel que fa a l'esquema de tercets **AAA-BBB** (i **AAA-BBB**), s'han representat totes dues variants (impossibilitat de reproduir en una columna aliena tant el tercet AAA com algun dels altres, BBB o CCC), però que la visió de l'esquema **AAA-AAA** (i **AAA-AAA**) ha estat incompleta. I tindria raó, perquè ens hem limitat a la impossibilitat de reproduir en una columna aliena els valors de AAA. Què passa amb els de BBB i CCC: doncs passa que no passa res, perquè <u>només podem assegurar que AAA apareix per tercera vegada</u>. A diferència del cas **1**

BBB AAA CCC
CCC **BBB** AAA
AAA CCC BBB, en què coneixent els 3 valors **AAA** i els 3 valors diferents **BBB** els 3 valors CCC queden determinats, en el cas **2**

BBB CCC AAA
CCC **AAA** BBB
AAA BBB CCC, conèixer els 3 valors **AAA** que es repetèixen no implica conèixer la composició de BBB i de CCC. Només constatàvem que els 3 valors CCC i els 3 valors BBB serien diferents (i, obviament, ho serien dels 3 valors **AAA**) i que s'anirien repetint (ordenació interna a part) per tot el requadre 9×3. Però amb els 6 valors 1... 9, exclosos els 3 de **AAA**, hi ha fins a 20 tercets possibles compatibles amb les altres caselles ocupades (fins a 10 parells de tercets), de manera que sempre serà possible agrupar en una columna aliena als dos tercets repetits no nom fins a 3 valors sinó fins a 4, tret d'haver-hi incompatibilitats amb altres ocupacions. Per tant, aquí teniu un exemple d'esquema **AAA** (esquerra) i **AAA** (dreta), amb tercet de valors diferents en columna aliena, i entre els dos la primera solució (comú, naturalment) trobada per **RE-MEMBER**:

0 0 0	0 0 0	0 0 0	9 6 4	5 7 3	8 2 1	0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	8 7 5	9 1 2	6 4 3	0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	3 1 2	6 8 4	9 7 5	0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	7 9 6	4 5 8	3 1 2	0 0 0	0 0 0	0 0 0
1 2 3	0 0 0	0 0 0	**1 2 3**	7 9 6	5 8 4	**1 2 3**	0 0 0	0 0 0
0 0 0	**3 2 1**	0 0 0	5 4 8	**3 2 1**	7 9 6	0 0 0	**3 2 1**	0 0 0
0 0 0	0 0 0	0 0 **7**	6 8 9	2 4 5	1 3 **7**	0 0 0	0 0 0	0 0 **7**
0 0 0	0 0 0	0 0 **8**	2 3 7	1 6 9	4 5 **8**	0 0 0	0 0 0	0 0 **8**
4 5 1	**8 3 7**	**2 6 9**	**4 5 1**	**8 3 7**	**2 6 9**	**4 5 1**	**8 3 7**	**2 6 9**

Així doncs, podríem simplificar **QUASI-PLE**, prescindint de la detecció de tercets **AAA-AAA**, i complicar les coses per un altra banda per tal d'afinar el perfil dels casos conflictius, sobreponderant els requadres 9×3 amb tercets **AAA-BBB** només en els casos en què la imminència d'un conflicte de compatibilitat ho fes necessari: en completar-se sobre una columna la presència de tres valors, algun dels quals hi

tingués compromesa una altra posició (cas dels tercets **AAA-BBB** en un requadre 9×3, amb l'emplenament d'una casella en algun dels dos requadres 3×3 aliens), o bé en completar-se sobre una fila un tercet que impliqués en determinada casella algun dels tres valors ja presents a la mateixa columna (cas dels tercets **AAA-BBB** en un requadre 9×3 i columna amb tres caselles plenes en els dos requadres 3×3 aliens).

Bona part dels esforços es van reorientar en aquesta direcció (donant lloc a unes andròmines tan inútils com inclassificables pel que fa a les estructures de dades perpetrades), que finalment fou abandonada. Què va passar per deixar-la de banda? Va passar que la norma de situar el més avall possible els requadres 9×3 integrats exclusivament per 3 tercets repetits no era aplicable quan això s'esdevenia en dos requadres alhora (quin dels dos hauríem de col·locar primer lloc?), ni tampoc quan els tercets explícits eren del tipus **AAA-AAA** (o **AAA-AAA**) i el mateix tercet sortia en dos requadres 9×3. En definitiva, podríem aprofitar en part els dos vessants de **QUASI-PLE** (detecció de configuracions **AAA-AAA** / **AAA-AAA** i **AAA-BBB** / **AAA-BBB**), però la cerca hauria de donar prioritat a un altre tipus de resultats, en el marc d'una estratègia diferent, vist (de seguida ho veurem) que això de falsejar l'ocupació per forçar la reordenació de l'escaquer no sempre funciona. Veiem-ne la inutilitat en tres exemples de configuracions **AAA-AAA** i **AAA-BBB** repetides en diferent mesura:

0 0 0	0 0 0	**1 4 7**	0 0 0	0 0 0	**1 4 7**	0 0 0	0 0 0	0 0 **7**
0 0 0	0 0 0	**2 5 8**	0 0 0	0 0 0	**2 5 8**	0 0 0	0 0 0	0 0 **8**
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	**3 6** 0	0 0 0	0 0 0	0 0 0
(789)	(456)	(123)	(789)	(456)	(123)	0 0 0	0 0 0	(789)
4 5 6	(123)	(789)	**4 5 6**	(123)	(789)	**8 9 7**	0 0 0	0 0 0
(123)	**7 8 9**	(456)	(123)	**7 8 9**	(456)	0 0 0	**7 8 9**	0 0 0
(456)	(789)	(123)	0 0 0	0 0 0	(789)	0 0 0	0 0 0	(789)
7 8 9	(123)	(456)	**7 8 9**	0 0 0	0 0 0	**7 8 9**	0 0 0	0 0 0
(123)	**4 5 6**	(789)	0 0 0	**8 9 7**	0 0 0	0 0 0	**8 9 7**	0 0 0

El primer i l'últim responen als casos esmentats a l'últim paràgraf, i el del mig en seria un d'intermedi. Però abans que res haurem de referir-nos a la notació que s'ha adoptat: en tots tres casos, les tres xifres sense separar i tancades dins de parèntesi representen els tres únics valors que poden ser assignats a la tríada de caselles, en l'ordre representat o no, i cada columna de valors dels requadres 3×3 superiors drets (l'últim, subratllat, segueix identificant la casella actual amb un candidat que serà refusat, però no abans d'una llarga espera) és, en realitat, una opció alternativa. Concretant: en el primer exemple, si sobre el requadre 3×9 aliè als tercets explícits 456 i 789 (123 implícit) anem acumulant en una mateixa columna valors constitutius d'un d'ells, ja no podrem introduir-ne dos com abans, perquè el segon (**2**, **5** o **8**) barraria el pas a un dels dos únics valors que resten (2 i 3, 5 i 6, o 8 i 9), i la prospecció mecànica d'aquests valors es farà llarga; en el tercer, tampoc no servirà de res que l'exploració d'un tercer valor amb 789 es tradueixi en la vulneració de les **normes 2** i **3** en considerar els requadres 3×3 central dret i inferior dret de la pseudomatriu **R-LLL**, perquè la llarga espera ja es produirà en l'exploració del segon (**8**) i per les mateixes raons d'abans, tot i que en aquest cas encara serà més dilatada (de seguida ho justificarem); pel que fa al segon, d'entrada suposarem que els temps d'espera que cal reduir se situen entre els altres dos i corresponen a la segona coincidència en columna (**8**) amb el tercet 789, present en dos requadres 9×3, i a la tercera (**6** i **3**) amb els tercets 456 i 123, només presents en un. Limitant-nos però als casos extrems, ¿per què hem dit que les durades crítiques (caselles 9,8, per desqualificar els valors **8** i **9**) encara eren més grans en l'exemple de la dreta que en el de l'esquerra? Dintre de poc quantificarem els retrocessos amb una de les versions-test utilitzades en el subcapítol precedent, però també ho podem justificar imaginant en ambdós casos la primera embranzida fins arribar a un cul-de-sac, que en el primer cas (esquerra) es produeix a la casella 9,5 i en el tercer (dreta) a la casella 9,6... (o almenys això és el que sembla: més endavant veurem que el tercer no arriba tan lluny), si reproduïm la dinàmica NATURMÍN i ometem les retirades de curt abast que es donen en el segon requadre 9×3 per ajustar l'emplenament d'algun tercet. Com hem vist en altres situacions on un veredicte negatiu es feia esperar massa, a partir d'aquest cul-de-sac es produeix una regressió zigzaguejant, en què cada retirada parteix del mateix punt (però arriba més enrera) i és seguida d'una nova embranzida, fins que arriba al primer tercet lliure i fa el cant del cigne en l'última embranzida. El cul-de-sac més avançat en l'últim exemple ja justificaria en ell mateix que el veredicte d'incompatibilitat trigués més, però encara hi ha un factor més decisiu:

0 0 0	0 0 0	0 0 **7**	A l'esquerra el tercet 1 pot	0 0 0	0 0 0	0 0 **7**
0 0 0	0 0 0	0 0 **8**	adoptar 6 formes, que són el	0 0 0	0 0 0	0 0 **8**
0 0 0	0 0 0	0 0 0	nombre de permutacions de 3	0 0 0	0 0 0	0 0 0
			elements (1, 2 i 3): 3! = 6.			
0 0 0	0 0 0	0 0 0		3 6 5	2 1 4	0 0 X
4 5 6	2 3 1	0 0 X	A la dreta el tercet 1 pot	**8 9 7**	3 6 5	2 1 4
2 3 1	**7 8 9**	5 6 4	adoptar 120 formes, que són	2 1 4	**7 8 9**	3 6 5
			el nombre de variacions de 6			
5 6 4	8 9 7	1 2 3	elements (1, 2, 3, 4, 5 i 6)	4 5 6	1 2 3	7 8 9
7 8 9	1 2 3	4 5 6	agrupats de tres en tres:	**7 8 9**	4 5 6	1 2 3
1 2 3	**4 5 6**	7 8 9	6 × 5 × 4 = 120.	1 2 3	**8 9 7**	4 5 6

En l'exemple de l'esquerra, el primer tercet queda limitat als valors 1, 2 i 3 pels seus veïns de fila (**4-5-6**) i requadre 3×3 (**7-8-9**), de manera que podem omplir les 3 caselles amb qualsevol de les 6 permutacions possibles amb aquests valors. El mateix grau de llibertat podem considerar en els tercets cinquè i novè (1, 2 i 3) i sisè (4, 5 i 6), però el tercer (7, 8 i 9) està condicionat per la presència dels valors **7** i **8** al final de les línies 8ª i 9ª, i només admet les 2 permutacions 7-8-9 i 8-7-9, i setè i vuitè estan condicionats per la presència de les tríades **4-5-6** i **7-8-9** en el segon requadre 9×3, respectivament, i només n'admeten 2 cada un: 5-6-4 i 6-4-5, i 8-9-7 i 9-7-8. De forma que en el primer requadre 9×3 hi ha $\mathbf{6^4 \times 2^3 = 10.368}$ combinacions possibles de tercets. En el segon requadre 9×3, la composició de les tríades també està determinada per la presència de dos tercets explícits (és la mateixa que en el primer requadre) però, com en els tres últims tercets esmentats, les possibilitats de permutació d'aquests valors sempre està limitada a 2 permutacions circulars a partir de l'existent en el primer requadre, per evitar coincidències en columna, de forma que les combinacions possibles de tercets són $\mathbf{2^3 = 8}$. Així doncs, a l'exemple de l'esquerra hi ha $\mathbf{10.368 \times 8 = 82.944}$ combinacions possibles de tercets (hauríem de parlar de combinacions d'emplenament de les caselles lliures però, ja que s'han calculat per tercets, donem-ho per bo), totes recorregudes perquè cap exploració no ha anat més enllà de la casella 6,5.

En l'exemple de la dreta, el primer tercet pot nodrir-se dels valors 1, 2, 3, 4, 5 i 6, perquè la composició dels veïns de fila i requadre 3×3 és idèntica (**7-8-9**), de manera que podem emplenar les 3 caselles amb qualsevol de les 120 variacions ternàries possibles amb aquests valors. Però, atenció!: un cop haguem optat per algun d'aquests 120 tercets (com 1-2-3), els tres valors escollits es repetiran en les altres dues aparicions del tercet en el primer requadre 9×3, i no admetran més canvi que les 6 permutacions possibles amb aquests valors. Pel que fa a les altres 4 tríades i anàlogament a l'exemple anterior, la seva composició està determinada, però així com en els dos altres tercets 123 i els tres 456 podem jugar lliurement amb les 6 permutacions, el tercet implícit 789 està condicionat per la presència dels valors **7** i **8** al final de les línies 8ª i 9ª, i només admet les permutacions 7-8-9 i 8-7-9. De forma que en el primer requadre 9×3 hi ha $\mathbf{120 \times 6^5 \times 2 = 1.866.240}$ combinacions possibles de tercets. Però en seguir amb el segon requadre 9×3 haurem d'anar en compte, perquè el seu primer tercet també pot nodrir-se dels valors 1, 2, 3, 4, 5 i 6, però no totes les 120 variacions de tres valors seran admissibles: només les compatibles amb els tercets 1-2-3 i 4-5-6 que ja figuren més avall. Si considerem que en aquestes 120 variacions ternàries hi ha 20×**1**, 20×**2**, 20×**3**, 20×**4**, 20×**5** i 20×**6** tant en 1ª posició com en 2ª i 3ª, entre les 118 variacions que resten després d'excloure 1-2-3 i 4-5-6, n'hi haurà

- 38 que comencin amb **1** o **4**.
- 24 més que tinguin al mig **2** o **5** (24 més 14 que ja s'havien comptat abans, 38).
- 16 més que acabin en **3** o **6** (16 més 22 que ja s'havien comptat abans, 38).

Per tant, de compatibles n'hi haurà $\mathbf{118 - (38 + 24 + 16) = 118 - 78 = 40}$. Però, igual com passava en el primer requadre 9×3, un cop n'haguem triat una (com 2-1-4), els tres valors es repetiran en les altres dues aparicions del tercet però no admetran més canvis que 2 permutacions (és a dir que, si a baix hi tenim 1-2-3 i 4-5-6, al damunt només s'hi pot posar 2-1-4 o 2-4-1) i el mateix pel que fa al tercer tercet (sols queden 3, 5 i 6 per assignar), de manera que en cadascuna de les seves tres aparicions també hi haurà 2 possibilitats (3-6-5 o 6-3-5) i, en conjunt, tindrem

40 × 2^5 = 1.280 combinacions de tercets. Així doncs, a l'exemple de la dreta hi ha **1.866.240 × 1.280 = 2.388.787.200** combinacions possibles d'emplenament de caselles lliures, totes recorregudes perquè cap exploració no ha anat més enllà de la 6,5.

Convè advertir al lector que els resultats parcials per requadres 9×3, en tots dos exemples, no són representatius de la quantitat de retrocessos que es produeixen a cadascun d'ells, perquè quan parlem de combinacions possibles pensem en nombre de permutacions que en total es produeixen en l'últim tercet emplenat virtualment, i això contempla el fet que les 2, 3 o 6 permutacions diferents de cada tercet es repeteixen tants cops com permutacions diferents hi ha en els tercets precedents. Per no fer-ho molt llarg i centrant-nos en el primer exemple diríem que: el primer tercet del primer requadre 9×3 canvia **6** vegades; l'últim tercet d'aquest mateix requadre canvia **10.368** vegades; el primer tercet del segon requadre 9×3 canvia **10.368 × 2 = 20.736** vegades, i l'últim tercet emplenat d'aquest mateix requadre canvia **82.944** vegades. Així, si per no haver de jugar amb mitjanes dubtosament representatives ens quedem amb els valors finals de cada requadre 9×3, diríem que:

- En el primer exemple:
  - En el primer requadre 9×3 s'han produït **10.368** permutacions.
  - En el segon requadre 9×3 s'han produït **10.368 × 8 = 82.944** permutacions.
- En el segon exemple:
  - En el primer requadre 9×3 s'han produït **1.866.240** permutacions.
  - En el segon requadre 9×3 s'han produït **1.866.240 × 1.280 = 2.388.787.200** permutacions.

Entre aquestes xifres i el nombre de retrocessos a cada requadre 9×3 sí que hi ha una certa correlació, i ho diem amb tanta prudència perquè, si d'una banda hem de considerar que cada permutació de valors en una tríada de caselles comporta un o més retrocessos, de l'altra hem de tenir ben present l'acció d'**ACTUALITZA-PLUS** < **ACTUALITZA** < **RE-MEMBER** emplenant caselles automàticament (acció que és més eficaç en el segon requadre que en el primer) que estalvia avanços i retrocessos.

Però no ens quedem amb les ganes de saber quants retrocessos hi ha exactament en cada cas, perquè l'eina ja la tenim (subcapítol precedent) i només caldrà disposar d'altes dosis de paciència per esperar els resultats:

- En el primer exemple:

```
     1>      0 +       0 +  0 =       0
     2>      0 +       0 +  0 =       0
     3>      0 +       0 +  0 =       0
     4>      0 +       0 +  9 =       9
     5>      0 +       0 +  0 =       0
     6>      0 +       0 +  6 =       6
     8< 18.921 + 145.152 +  0 = 164.073        10.368            82.944
     9< 18.921 + 145.152 +  0 = 164.073
     T> 37.842 + 290.304 + 15 = 328.161
```

- En el segon exemple:

```
     1>        48 +  0 + 0 =        48
     2>        48 +  2 + 0 =        50
     3>        48 +  2 + 0 =        50
     4>       126 +  0 + 1 =       127
     5>       126 +  0 + 0 =       126
     6>       126 +  6 + 1 =       133
     8< 3.712.116 +  0 + 0 = 3.712.116      1.866.240     2.388.787.200
     9< 3.712.116 +  0 + 0 = 3.712.116
     T> 7.424.754 + 10 + 2 = 7.424.766
```

Els resultats del primer exemple són perfectament plausibles, però els del segon no poden ser més desconcertants, i no tant per l'elevadíssim nombre de retrocessos en el primer requadre 9×3 sinó per la total absència en el segon: ¿que no havíem quedat que les embranzides arribaven fins al seu penúltim tercet? Doncs si, però aquí la intervenció d'**ACTUALITZA-PLUS** < **ACTUALITZA** < **RE-MEMBER** emplenant caselles automàticament ha anat més lluny que no ens esperàvem: com veureu en la seqüència representada, quan encara **RE-MEMBER** estava a la meitat de l'emplenament virtual previst per a l'examen del candidat **8** (9,8), just després de posar 7 a la casella 7,3 (esquerra), **ACTUALITZA-PLUS** col·loca d'ofici un altre 7 a 8,6, un 8 a 7,6 i un 9 a 9,6 (centre), de manera que la casella 9,3 es queda sense candidats (dreta).

```
0 0 0   0 0 0   0 0 7     0 0 0   0 0 0   0 0 7     0 0 0   0 0 0   0 0 7
0 0 0   0 0 0   0 0 8     0 0 0   0 0 0   0 0 8     0 0 0   0 0 0   0 0 8
0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   8 7 9     0 0 0   0 0 0   8 7 9
8 9 7   0 0 0   0 0 0     8 9 7   0 0 0   0 0 0     8 9 7   0 0 0   0 0 0
0 0 0   7 8 9   0 0 0     0 0 0   7 8 9   0 0 0     0 0 0   7 8 9   0 0 0

4 5 6   1 2 3   7 0 0     4 5 6   1 2 3   7 0 0     4 5 6   1 2 3   7 0 X
7 8 9   4 5 6   1 2 3     7 8 9   4 5 6   1 2 3     7 8 9   4 5 6   1 2 3
1 2 3   8 9 7   4 5 6     1 2 3   8 9 7   4 5 6     1 2 3   8 9 7   4 5 6
```

Per la casella 7,3 encara desfilaran els valors 8 i 9: després del primer, tornarà a intervenir-hi **ACTUALITZA-PLUS**, col·locant d'ofici un altre 8 a 8,6, un 7 a 7,6 i un 9 a 9,6, i 9,3 es quedarà en punt mort; pel que fa al segon, no li caldrà cap mitjancer per deixar 9,3 en la mateixa situació. I des d'aquest punt es posarà en marxa una regressió zigzaguejant fins a l'inici, en què cada embranzida arrencarà d'una posició anterior i inexorablement acabarà estavellant-se en el mateix lloc. Ens hem molestat a descriure amb detall l'inici de la retirada, amb l'oportuna i reiterada intervenció d'**ACTUALITZA-PLUS**, perquè el lector s'adoni de com n'és de lenta i no li vinguin massa de nou els **3.712.116** retrocessos. Val a dir que això de "oportuna" no és broma, perquè si no fos per la inclusió d'**ACTUALITZA-PLUS** < **ACTUALITZA** a **RE-MEMBER**, imagineu-vos com seria una regressió que no s'iniciés fins la posició 9,6, com havíem previst. De moment, una cosa més que hem aprés és que, de vegades, hi ha factors que actuen en el sentit d'escurçar els temps d'espera, tot i que no sempre n'hi ha prou amb aquesta contribució favorable per poder-los homologar. No podem tancar l'anàlisi d'aquests dos casos sense abans validar unes estimacions obtingudes a partir de pressupostos teòrics que la via empírica s'ha encarregat de desmentir: havent resituat el punt mort al final del primer requadre 9×3, ja no podem seguir dient que en el segon cas hi ha **2.388.787.200** combinacions possibles de tercets; cal actualitzar el còmput a $\mathbf{120 \times 6^5 \times 3 = 2.799.360}$ (l'últim factor **3** és el nombre d'intents avortats a l'últim tercet d'aquest primer requadre 9×3), valor més en consonància amb els **3.712.116** retrocessos (no ens oblidem que el nombre de retrocessos és superior al de possibilitats d'emplenament). Tot i que encara té més transcendència pràctica haver constatat que, si el recurs de situar en primera posició el requadre 9×3 conflictiu no sempre serveix, no és sols perquè quan n'hi ha dos afavorir-ne un comporta castigar l'altre, sinó perquè en última instància és el nombre de combinacions de valor del primer tercet lliure allò que determina que la regressió sigui més o menys llarga: en aquest últim exemple, el segon requadre 9×3 contribueix a desfermar el conflicte i tanmateix la regressió fins al tercet inicial es circumscriu al primer, però el que la fa insuportable és que no acabarà fins que en primera posició no s'instal·li la combinació de valors 6-5-4 (del sextet 1-2-3-4-5-6), que donarà peu a l'última embranzida fracassada.

És precisament el fet que l'última embranzida s'inicii amb la permutació 3-2-1 del tercet 1-2-3, i no pas amb 6-5-4, allò que determina la volada més curta (espera llarga, però no tant) del primer dels tres exemples representats quatre pàgines enrera: la primera es representava tres pàgines enrera i, a diferència del tercer exemple (la primera embranzida del qual també s'hi representava, a la dreta, però no reflectia l'exploració real abreujada per la intervenció d'**ACTUALITZA-PLUS**), sí que responia a l'exploració. D'aquests tres exemples, els dos primers admetien una variant amb els dos primers requadres 9×3 permutats i potser algun lector quedaria frustrat si no pogués comparar-ne l'abast de la volada amb les representacions que ja hem ofert, així que no quantificarem temps d'espera però mostrarem la primera embranzida dels casos que quedaven per considerar i fins i tot gosarem avançar els resultats comparatius, qualitativament: del primer exemple en presentem la variant (esquerra) de l'exposada tres pàgines enrera, i del segon, del qual havíem passat en aquella ocasió, en presentarem les dues (centre i dreta). Pel que fa al primer cas, és raonable suposar que l'espera serà menor en el representat ara que en el vist tres pàgines enrera, atès que llavors les embranzides arribaven a la 5ª fila i ara a la 4ª. Quant a les dues variants del segon exemple, l'embranzida inicial de les quals representem a continuació, no hi ha dubte que la primera suposa una exploració més llarga que la segona (la més curta de les cinc considerades, perquè les embranzides es queden a la 2ª fila, bloquejades per l'emplenament 9,6 (9) que

ACTUALITZA-PLUS afegeix després d'ocupar 7,2 amb 7 i 8, i directament en ocupar-la amb 9) i també que la del tercer exemple, la fins ara més llarga: no només perquè les embranzides arriben fins a la 5ª fila, en comptes de quedar-se a la 3ª, sinó perquè l'última embranzida arrencarà amb la combinació de valors 6-5-4 instal·lada en el tercet inicial, igual com passava en aquest tercer exemple. Així doncs, allò que havíem suposat quatre pàgines enrera (que els temps d'espera se situaven entre els del primer i tercer exemple) serà ben bé al contrari: depenent de la posició relativa de les configuracions **AAA-AAA** i **AAA-BBB**, comportarà l'espera més llarga (més llarga que el tercer exemple) o la més curta (més curta que el primer).

```
0 0 0   0 0 0   0 0 7     0 0 0   0 0 0   0 0 7     0 0 0   0 0 0   0 0 7
0 0 0   0 0 0   0 0 8     0 0 0   0 0 0   0 0 8     0 0 0   0 0 0   0 0 8
0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   8 7 9
7 8 9   0 0 0   0 0 0     4 5 6   2 3 1   0 0 X     7 8 9   0 0 0   0 0 0
2 3 1   4 5 6   0 0 X     2 3 1   7 8 9   5 6 4     0 0 0   8 9 7   0 0 0

8 9 7   5 6 4   1 2 3     5 6 4   1 2 3   7 8 9     0 0 0   0 0 0   0 0 0
4 5 6   1 2 3   7 8 9     7 8 9   4 5 6   1 2 3     4 5 6   1 2 3   7 0 X
1 2 3   7 8 9   4 5 6     1 2 3   8 9 7   4 5 6     1 2 3   7 8 9   4 5 6
```

Com que l'autor ja havia hagut de suportar una espera de quinze hores per arribar a completar el recompte de **7.424.766** retrocessos consignats dues pàgines enrera i no li han quedat ganes d'exposar-se a una altra espera del mateix ordre, el lector suspicaç és convidat a provar-ho després de practicar unes petites modificacions en les funcions **INFORMA** i **MASCARA**: n'hi haurà prou amb la substitució del **0** enter pel **0.0** real en les inicialitzacions de les variables *ad hoc* (***REQ-1***, ***REQ-2***, ***REQ-3***, ***TOT-1***, ***TOT-2*** i ***TOT-3***) d'aquesta versió-test, substituint també les expressions **(itoa (if TOT ...))** de la primera funció per **(rtos (if TOT ...) 2 0)**.

Ens hem esplaiat prou veient com la reordenació de requadres 9×3 no és la panacea, perquè en molts casos no serveix de res. També hem deixat anotat que, fins adonar-nos d'això, l'última versió mostrada es va anar transformant en un trasto cada cop més inflat i més inútil, i deslliurarem el lector del càstig de conèixer el cúmul de despropòsits comesos. Alternativament, presentarem les circumstàncies que van contribuir a orientar la resolució del problema en la direcció correcta: que si la presència de configuracions **AAA-AAA** i **AAA-BBB** (per no fer-ho tan llarg, a partir d'ara quan les designem així entendrem que també ens referim a **AAA-AAA i AAA-BBB**) permet simplificar els elements de **0-LLL** (**R-LLL**, dins de **RE-MEMBER**) en el requadre 9×3 corresponent, reduint a tres el nombre de candidats admissibles (el nombre de valors no nuls d'aquests elements) en les caselles desocupades d'alguns o de tots els seus tercets, tal i com havíem representat entre parèntesis en la pàgina 254 (com que en aquelles representacions els candidats ocupaven l'espai d'un tercet i no podíem repetir-los per a cada casella, ja enteníem que eren candidats comuns), sols cal fer això, i la repercusió sobre la durada de l'exploració serà immediata.

Tan capficat estava l'autor amb la falsa premissa que tots aquests mals havien de respondre positivament a la teràpia d'alguna reordenació miraculosa dels requadres 9×3, i que només faltava precisar com s'havien de ponderar les ocupacions reals o potencials, segons que responguessin a la configuració **AAA-AAA** o a la **AAA-BBB**, que l'evidència del que s'acaba d'afirmar no va brillar per ella mateixa, sinó que li va caldre esperar que un dispositiu provisional de treball concebut justament per no haver de suportar llargues esperes en la realització de diverses comprovacions, el fes prendre consciència que, sense haver-s'ho proposat, estaba en el bon camí. D'entrada, no considerava que aquesta mena de lubricant fos res més que un recurs auxiliar, per la simple raó que la seva producció no estava programada i era pura artesania: cada cop calia fabricar una llista d'associacions **33VV** a mida del cas que es volia analitzar. **33VV** no pretenia ser altra cosa que una amalgama *ad hoc* de la informació continguda a **R-9*9** i **R-LLL**, limitada a les posicions dels tercets implícits de les configuracions **AAA-AAA** i **AAA-BBB** (pel que fa a **R-9*9**) i als tres candidats comuns (pel que fa a **R-LLL**), on a cada subllista les coordenades de la casella era l'element-clau i el segon element era el seu homòleg a **R-LLL**, del qual algun *deus ex machina* (el pacient programador, en una assignació prèvia) havia fet desaparèixer els valors nuls i redundants. Per acabar-ho d'entendre (sense cap altre propòsit que copsar el grau d'obcecació de l'autor), haurem d'indicar que la variable **33VV** era local a **OMPLE-SUDOKU**, se li assignava valor a **MASCARA** i s'usava a l'inici de **RE-MEMBER**, inici que reproduïm seguint dues funcions subordinades:

```
(defun VCOMUNS (/ VC)
  (foreach V (reverse NV)
    (setq VC (cons (if (and V (member V 3V)) V) VC))))

(defun SUBST-LLL (/ X Y F 3V NV LLL)
  (foreach Y '(8 7 6 5 4 3 2 1 0)
    (setq F ())
    (foreach X '(8 7 6 5 4 3 2 1 0)
      (setq 3V (cdr (assoc (list X Y) 33VV)) NV (nth X (nth Y R-LLL))
            F (cons (if 3V (VCOMUNS) NV) F)))
    (setq LLL (cons F LLL))))

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI
      (progn
        (if 33VV (setq R-LLL (SUBST-LLL)))
        (setq 2R (ACTUALITZA R-9*9 N (if E (last (last E)) P) R-LLL ())
              INI (EXPLORA-3*9) R-9*9 (car 2R) R-LLL (cadr 2R)))
  (if (not (or INI OK)) ...)
  OK)
```

Per a aquestes comprovacions o s'aprofitava una versió antiga de **MASCARA** en què ni ****9*9**** ni **LLL** ni **P** no s'haguessin "optimitzat" (és a dir, en què no es practiqués cap transposició ni permutació) o s'anulaven temporalment aquestes modificacions amb un afegitó curt i fàcil de suprimir, si s'aprofitava la penúltima versió, o a més d'això es neutralitzava ***PP*** per evitar el tractament reservat als tercets en potència **AAA**-**AAA** o **AAA**-**BBB**, si el present dispositiu de treball (que no contempla aquesta possibilitat) es muntava directament sobre l'última versió. Si, dels tres exemples que considerem tot seguit (demanant disculpes perquè aquest "tot seguit" ens traspassi les representacions a la pàgina següent) i que tenen en comú que el tercet 789 queda implícit a dos dels requadres 3×3 drets, prenem el de l'esquerra, la tercera de les modalitats descrites per construir la **MASCARA** provisional seria:

```
(defun MASCARA (L / INI OK JJ **T*T** 0-**9*9** 0-9 0-LLL 0-L 0-9*9 0-LL
                    0-T*T 0-TT TN TQ A TA NN TNN AA TAA *P* *P P* *PP*)
  (setq LL () SS (ssadd))
  (if POST
      (progn
        ....................................................
        (setq 0-**9*9** **9*9** 0-LLL LLL P (list X Y) *PP* ())
        (if (equal P '(8 8))
            (setq 33VV (list (cons '(6 2) '(7 8 9)) (cons '(7 2) '(7 8 9))
                             (cons '(8 2) '(7 8 9))
                             (cons '(6 5) '(7 8 9)) (cons '(7 5) '(7 8 9))
                             (cons '(8 5) '(7 8 9)))))))
  (foreach N L1A9 ...)
  (if POST ...))
```

Per a l'exemple del mig l'assignació de valor a **33VV**, en el mateix context, seria:

```
(setq 33VV
      (list (cons '(0 0) '(7 8 9)) (cons '(1 0) '(7 8 9)) (cons '(2 0) '(7 8 9))
            (cons '(6 0) '(1 2 3)) (cons '(7 0) '(1 2 3)) (cons '(8 0) '(1 2 3))
            (cons '(3 1) '(7 8 9)) (cons '(4 1) '(7 8 9)) (cons '(5 1) '(7 8 9))
            (cons '(6 1) '(4 5 6)) (cons '(7 1) '(4 5 6)) (cons '(8 1) '(4 5 6))
            (cons '(0 2) '(4 5 6)) (cons '(1 2) '(4 5 6)) (cons '(2 2) '(4 5 6))
            (cons '(3 2) '(1 2 3)) (cons '(4 2) '(1 2 3)) (cons '(5 2) '(1 2 3))
            (cons '(6 2) '(7 8 9)) (cons '(7 2) '(7 8 9)) (cons '(8 2) '(7 8 9))
            (cons '(0 3) '(7 8 9)) (cons '(1 3) '(7 8 9)) (cons '(2 3) '(7 8 9))
            (cons '(6 3) '(1 2 3)) (cons '(7 3) '(1 2 3)) (cons '(8 3) '(1 2 3))
            (cons '(3 4) '(7 8 9)) (cons '(4 4) '(7 8 9)) (cons '(5 4) '(7 8 9))
            (cons '(6 4) '(4 5 6)) (cons '(7 4) '(4 5 6)) (cons '(8 4) '(4 5 6))
            (cons '(0 5) '(4 5 6)) (cons '(1 5) '(4 5 6)) (cons '(2 5) '(4 5 6))
            (cons '(3 5) '(1 2 3)) (cons '(4 5) '(1 2 3)) (cons '(5 5) '(1 2 3))
            (cons '(6 5) '(7 8 9)) (cons '(7 5) '(7 8 9)) (cons '(8 5) '(7 8 9))))
```

I, per al de la dreta, seria:

```
(setq 33VV
      (list (cons '(0 0) '(7 8 9)) (cons '(1 0) '(7 8 9)) (cons '(2 0) '(7 8 9))
            (cons '(6 0) '(1 4 5)) (cons '(7 0) '(1 4 5)) (cons '(8 0) '(1 4 5))
            (cons '(3 1) '(7 8 9)) (cons '(4 1) '(7 8 9)) (cons '(5 1) '(7 8 9))
            (cons '(6 1) '(2 3 6)) (cons '(7 1) '(2 3 6)) (cons '(8 1) '(2 3 6))
            (cons '(0 2) '(2 3 6)) (cons '(1 2) '(2 3 6)) (cons '(2 2) '(2 3 6))
            (cons '(3 2) '(1 4 5)) (cons '(4 2) '(1 4 5)) (cons '(5 2) '(1 4 5))
            (cons '(6 2) '(7 8 9)) (cons '(7 2) '(7 8 9)) (cons '(8 2) '(7 8 9))
            (cons '(0 3) '(7 8 9)) (cons '(1 3) '(7 8 9)) (cons '(2 3) '(7 8 9))
            (cons '(6 3) '(1 2 3)) (cons '(7 3) '(1 2 3)) (cons '(8 3) '(1 2 3))
            (cons '(3 4) '(7 8 9)) (cons '(4 4) '(7 8 9)) (cons '(5 4) '(7 8 9))
            (cons '(6 4) '(4 5 6)) (cons '(7 4) '(4 5 6)) (cons '(8 4) '(4 5 6))
            (cons '(0 5) '(4 5 6)) (cons '(1 5) '(4 5 6)) (cons '(2 5) '(4 5 6))
            (cons '(3 5) '(1 2 3)) (cons '(4 5) '(1 2 3)) (cons '(5 5) '(1 2 3))
            (cons '(6 5) '(7 8 9)) (cons '(7 5) '(7 8 9)) (cons '(8 5) '(7 8 9))))
```

```
0 0 0   0 0 0   0 0 7       0 0 0   0 0 0   0 0 7       0 0 0   0 0 0   0 0 7
0 0 0   0 0 0   0 0 8       0 0 0   0 0 0   0 0 8       0 0 0   0 0 0   0 0 8
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   (789)       (456)   (123)   (789)       (456)   (123)   (789)
8 9 7   0 0 0   0 0 0       2 3 1   (789)   (456)       2 3 1   (789)   (456)
0 0 0   7 8 9   0 0 0       (789)   5 6 4   (123)       (789)   5 6 4   (123)

0 0 0   0 0 0   (789)       (456)   (123)   (789)       (236)   (145)   (789)
7 8 9   0 0 0   0 0 0       1 2 3   (789)   (456)       1 4 5   (789)   (236)
0 0 0   8 9 7   0 0 0       (789)   4 5 6   (123)       (789)   2 3 6   (145)
```

En tots tres exemples es tractava d'emplenar els tercets dels dos requadres 9×3 inferiors, i després passar a la casella 9,9 (**7**) i 9,8 (**8**). Amb l'emplenament de la penúltima casella, **33VV** adquiriria el valor previst i a partir d'aquest moment només els valors comuns a les llistes de **33VV** i **R-LLL** corresponents a la mateixa posició subsistirien en les de la segona variable. La idea de l'autor era que amb aquest dispositiu la longitud de l'exploració disminuiria, els temps d'espera dels candidats conflictius serien més assumibles i podria comparar a cadascun d'aquests i d'altres exemples els temps resultants de diverses disposicions relatives dels requadres 9×3 amb configuracions **AAA-AAA** i **AAA-BBB** (naturalment, per a cada nova posició caldria canviar en **MASCARA** no només les coordenades-clau de les subllistes de **33VV** sinó la casella **P** que en anar a emplenar-se posa en marxa la simplificació de **R-LLL**, atès que cal traslladar els emplenaments crítics al tercer requadre). I va ser la constatació que les esperes no només s'escurçaven sinó que desapareixien el que a l'autor li va fer saltar la bena dels ulls: les eventuals transposicions i reordenacions de requadres 9×3 i files ja no podien donar més de si, i l'únic que podia salvar la situació en presència de configuracions **AAA-AAA** i **AAA-BBB** (o **AAA-AAA** i **AAA-BBB**, que amb el nou enfoc ja no calia diferenciar operativament de les anteriors) era l'explicitació dels tercets implícits, mitjançant la depuració de **R-LLL**. Al capdavall, <u>només</u> calia aconseguir que allò que sabem nosaltres i ens ha permés de fabricar **33VV**, ho sàpiga també el programa: que no es quedi amb la simple detecció d'aquestes configuracions singulars, cosa que ja feia **QUASI-PLE**, sinó que d'això en pugui deduir la ubicació i composició dels tercets implícits. El que passa és que hem de desplaçar l'èmfasi des de la localització de tercets de valors assignats (**R-9*9**) a la de tercets de valors assignables (**R-LLL**), perquè no sempre aquests darrers es deriven de les configuracions descrites. Veieu, si no, com els tres casos següents presenten tercets 789 implícits en dos dels requadres 3×3 drets, mentre que únicament el primer respon a la típica configuració **AAA-AAA**:

```
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0

0 0 0   0 0 0   (789)       0 0 0   0 0 0   (789)       0 0 0   0 0 0   (789)
8 9 7   0 0 0   0 0 0       8 9 0   0 7 0   0 0 0       8 9 0   0 7 0   0 0 0
0 0 0   8 9 7   0 0 0       0 7 0   8 9 0   0 0 0       0 7 0   8 9 0   0 0 0

0 0 0   0 0 0   (789)       0 0 0   0 0 0   (789)       0 0 0   0 0 0   (789)
7 8 9   0 0 0   0 0 0       7 8 9   0 0 0   0 0 0       0 8 9   0 0 7   0 0 0
0 0 0   7 8 9   0 0 0       0 0 0   7 8 9   0 0 0       0 0 7   0 8 9   0 0 0
```

També els tres nous casos presenten idèntics tercets implícits al final de les sis primeres files, mentre que sols el primer respon a la típica configuració **AAA-BBB**:

0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
(123)	(456)	(789)	(123)	(456)	(789)	(256)	(134)	(789)
4 5 6	(789)	(123)	**4 5 6**	**8 9 7**	(123)	**7 8 4**	**9 5 6**	(123)
(789)	**1 2 3**	(456)	**8 9 7**	**1 2 3**	(456)	**9 3 1**	**8 2 7**	(456)
(456)	(123)	(789)	(456)	(123)	(789)	(134)	(256)	(789)
1 2 3	(789)	(456)	**1 2 3**	**7 8 9**	(456)	**8 2 7**	**1 9 3**	(456)
(789)	**4 5 6**	(123)	**7 8 9**	**4 5 6**	(123)	**5 6 9**	**7 8 4**	(123)

Malgrat això, seguirem referint-nos a configuracions **AAA-AAA**, **AAA-BBB**, **AAA-AAA** i **AAA-BBB**, per no embolicar encara més la troca amb noves denominacions: pot ser que aquestes no siguin prou afortunades, però ja les tenim assumides i tampoc no costa massa que a partir d'ara els atribuïm un significat més ampli. I, ja que estem amb precisions semàntiques, serà prudent advertir al lector sobre certes llibertats que ens hem pres i seguirem prenent-nos a l'hora de parlar de "tercets implícits", "explicitació de tercets implícits" i "tercets explícits", expressions l'ús de les quals es presta a equívocs i que caldrà situar adequadament en funció del context. Quan parlem de tercets explícits ens referim a elements de **R-9*9**: valors assignats a 3 caselles d'un requadre 3×3 alineades horitzontalment o vertical; per exemple, **7, 8** i **9**. Quan parlem de tercets implícits ens referim a elements de **R-LLL**: valors potencials, valors admissibles o candidats (les tres denominacions les fem servir indiscriminadament) per a 3 caselles d'un requadre 3×3 alineades horitzontalment o vertical, però en alguns casos les llistes (en què seguim representant **nil** amb 0, perquè no ocupin una longitud excessiva) encara mantenen una informació obsoleta, com ara (1 2 0 4 0 0 **7 8 9**), (1 2 0 4 0 6 **7 8 9**) i (1 2 3 0 0 0 **7 8 9**), mentre que a d'altres ja tenim (0 0 0 0 0 0 **7 8 9**), (0 0 0 0 0 0 **7 8 9**) i (0 0 0 0 0 0 **7 8 9**) perquè han experimentat un procés de depuració del qual ens ocuparem de seguida; és aquesta versió depurada del tercet que representem simplificadament amb (789). I, com probablement haureu endevinat, quan parlem d'explicitar tercets implícits ens referim al procés depurador que converteix les llistes del primer tipus (on la brossa encara no deixa veure els 3 únics valors no nuls, comuns a tot el tercet) al segon (on els tercets implícits ja "s'han explicitat"), no en valors assignats a caselles, com podria suggerir l'associació precipitada amb "tercets explícits".

Però abans d'anar per feina i ja que amb els tres exemples anteriors a aquests sis no preteníem fer estudis tipològics però ho semblava, pel que fa als dos requadres 9×3 inferiors (en el primer, ambdós responien a configuracions **AAA-AAA** definides pel mateix tercet; en el segon, a configuracions **AAA-BBB** definides pels mateixos tres tercets; en el tercer, a configuracions **AAA-BBB** en què un tercet es repetia i els altres dos no), ens permetrem una última digressió. Passant d'escaquers 9×9 prou poblats com per definir configuracions **AAA-BBB** a SUDOKUs-SOLUCIÓ en què un, dos o els tres requadres 9×3 estan configurats així (ara tots els tercets seran explícits, és clar), podríem recordar-nos de la solució CANÒNICA (esquerra), per adonar-nos que tots tres requadres 9×3 són **AAA-BBB**, per bé que no tenen cap tercet en comú, i de les solucions NATURMÍN (centre) i NATURMAX, en què també tots tres requadres 9×3 són **AAA-BBB** però tenen un tercet comú (789 i 123, respectivament):

9 1 2	3 4 5	6 7 8	9 7 8	5 3 1	6 4 2	9 7 8	3 1 2	6 4 5
6 7 8	9 1 2	3 4 5	6 4 2	9 7 8	5 3 1	6 4 5	9 7 8	3 1 2
3 4 5	6 7 8	9 1 2	5 3 1	6 4 2	9 7 8	3 1 2	6 4 5	9 7 8
8 9 1	2 3 4	5 6 7	8 9 7	2 1 4	3 6 5	8 9 7	2 3 1	5 6 4
5 6 7	8 9 1	2 3 4	3 6 5	8 9 7	2 1 4	5 6 4	8 9 7	2 3 1
2 3 4	5 6 7	8 9 1	2 1 4	3 6 5	8 9 7	2 3 1	5 6 4	8 9 7
7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9	1 2 3	4 5 6	7 8 9
Solució CANÒNICA			Solució NATURMÍN			Solució PANÒNICA		

En realitat, quan en el subcapítol precedent remenàvem configuracions pròximes a les solucions CANÒNICA, NATURMÍN i NATURMAX, vam estar a un pas de topar amb la problemàtica que ara tractem, i si no ens va afectar va ser per pura casualitat. I, ja que estem considerant aquestes velles conegudes des d'una altra perspectiva, podríem completar el repertori amb una variant de les dues solucions representades (dreta) en què els dos requadres 9×3 superiors fossin rèpliques de l'inferior pel que fa a la composició dels tercets, que es permuten circularment per mantenir la compatibilitat en columna. Com a petit homenatge personal a la baronessa Pannonica de Koenigswarter (per les barbes de Sant Celoni Monk, tingueu cura de no escriure Pannonica *von* Koenigswarter) i per la seva similitud amb la CANÒNICA (les columnes 1, 4 i 7 són idèntiques i la resta tenen igual composició per tercets), recuperem del capítol 2 la solució PANÒNICA (dreta), que més endavant tornarem a utilitzar per aproximar-nos a la qüestió de si hi ha algun tipus de SUDOKU-SOLUCIÓ amb més predisposició que les altres a l'obtenció de SUDOKUs-PROBLEMA amb baixa ocupació.

Dit això, posem fil a l'agulla exposant les bases en què es fonamenta l'estratègia de localització de tercets implícits (recordem-ho: parlem de tercet implícit quan els elements de **R-LLL** corresponents a tres caselles d'un requadre 3×3 alineades en fila o columna són llistes idèntiques amb només tres membres no nuls, membres que coincideixen amb els tres únics candidats admissibles). Per no haver d'utilitzar símbols alfabètics que podríem confondre amb altres conceptes, recorrerem a valors concrets però amb el benentès que la funció denotativa és genèrica: quan parlem de **1**, **2** i **3** podríem referir-nos a tres nombres **1**, **2**... **9** qualssevol.

Pel que fa a configuracions **AAA-AAA**, la identificació de tercets implícits serà possible gràcies al postulat **A** següent:
- Si en dos tercets d'un requadre 3×3 ens trobem que:
  - Les caselles ocupades (si n'hi ha) ho són amb valors diferents de **1**, **2** i **3**.
  - Les caselles lliures (si n'hi ha) no admeten cap dels valors **1**, **2** i **3**.
- Llavors s'esdevé en el tercer tercet que:
  - Les caselles ocupades (si n'hi ha) ho seran amb els valors **1**, **2** o **3**.
  - Les caselles lliures (si n'hi ha) només admetran valors **1**, **2** i **3** (els que no figurin entre els precedents).

Fixeu-vos que, en els exemples representats al final de la penúltima pàgina, **A** és aplicable a tots tres i no únicament al primer (allà tenim **7**, **8** i **9**).

Pel que fa a configuracions **AAA-BBB**, la identificació de tercets implícits serà possible (tot i que en determinats casos caldrà recórrer prèviament al postulat **A**) gràcies als postulats **B** i **C** següents:
- Si en un tercet d'un requadre 3×3 (**B**) o en una fila/columna (**C**) ens trobem que:
  - Les caselles ocupades (si n'hi ha) ho són amb els valors **1**, **2** o **3**.
  - Les caselles lliures (si n'hi ha) només admeten valors **1**, **2** i **3** (els que no figurin entre els precedents).
- Si en un altre tercet ens trobem que:
  - Les caselles ocupades (si n'hi ha) ho són amb valors diferents de **1**, **2** i **3**.
  - Les caselles lliures (si n'hi ha) només admeten valors diferents de **1**, **2** i **3** (els que no figurin entre els precedents).
  - Entre els valors de les caselles ocupades i els que admeten les lliures, dels sis valors diferents de **1**, **2** i **3** només en surten tres: **4**, **5** i **6**.
- Llavors s'esdevé en el tercer tercet que:
  - Les caselles ocupades (si n'hi ha) ho seran amb els valors **7**, **8** o **9**, diferents dels anteriors.
  - Les caselles lliures (si n'hi ha) només admetran valors **7**, **8** i **9** (els que no figurin entre els precedents).

Fixeu-vos que, en els exemples representats a l'inici de l'última pàgina, **B** es pot aplicar a tots tres i no únicament al primer (amb valors diversos a cada exemple). En el cas concret que les sis caselles dels dos primers tercets estiguin ocupades (per **1**, **2**, **3**, **4**, **5** i **6**), no caldrà instrumentar cap dispositiu especial, sinó que **ACT-LLL** ja haurà deixat les caselles lliures del tercer tercet sols amb els valors admissibles (de **7**, **8** i **9**, els que no ocupin cap casella en el tercet).

Si a la notació que usàvem darrerament (xifres en negreta per als valors assignats i xifres sense separar entre parèntesi per als tres únics valors admissibles dels tercets implícits) li afegim ~~xifres ratllades~~ entre parèntesi per indicar que en cap de les caselles del tercet no són admissibles aquests valors, la seqüència de postulats que haurem d'aplicar en cada cas fins arribar a explicitar un (**AAA-AAA**) o tres (**AAA-BBB**) tercets implícits, la podem esquematitzar d'aquesta manera:

```
(~~123~~)   1 2 3   (~~123~~)       (~~123~~)   1 2 3   (~~123~~)
1 2 3       (~~123~~)   (~~123~~)   1 2 3       (~~123~~)   (~~123~~)
(~~123~~)   (~~123~~)   0 0 0       (~~123~~)   (~~123~~)   (123)
                        A

7 8 9   4 5 6   (123)       7 8 9   4 5 6   (123)
1 2 3   7 8 9   (456)   ->  1 2 3   7 8 9   (456)
(456)   (123)   0 0 0       (456)   (123)   (789)
                B

(~~123~~)   1 2 3   (~~123~~)       (~~123~~)   1 2 3   (~~123~~)       (~~123~~)   1 2 3   (456)
1 2 3       4 5 6   (789)   ->  1 2 3       4 5 6   (789)   ->  1 2 3       4 5 6   (789)   ->
(~~123~~)   (789)   0 0 0       (~~123~~)   (789)   (123)       (~~123~~)   (789)   (123)
                    A                               B               C

                                (789)   1 2 3   (456)
                            ->  1 2 3   4 5 6   (789)
                                (456)   (789)   (123)

0 0 0   0 0 0   0 0 0       (789)   (123)   0 0 0       (789)   (123)   (456)
1 2 3   (456)   0 0 0   ->  1 2 3   (456)   0 0 0   ->  1 2 3   (456)   (789)
(456)   7 8 9   0 0 0       (456)   7 8 9   0 0 0       (456)   7 8 9   (123)
  B       B                                   C
```

Aprofitant l'estructura d'exploració per requadres 3×3 d'**EXPLORA-3*9**, la funció **TERCETS** detectarà les situacions **A** (1>(no 123) + 2>(no 123) -> 3>(només 123)) i **B** (1>(només 123) + 2>(només 456) -> 3>(només 789)). Un cop desvetllada la naturalesa de tercets complementaris d'aquests requadres 3×3, identificar-ne el tercer en un requadre 9×3 o 3×9 només ho podrem fer amb la funció **FILS/COLS**, que detectarà **C**. Hem d'aclarir que **TERCETS** disposa d'un senyal **SEGUIR**, que s'activa en intervenir **A** i provoca que se'n reiteri l'execució, per detectar si després d'aquesta depuració és aplicable **B**: si no, en el tercer esquema no s'efectuaria la segona depuració.

Si **TERCETS** hagués d'usar com a patró de comparació les **84** combinacions ternaries possibles entre els 9 valors 1... 9, trigaria tres vegades més que usant les només **28** combinacions que inclouen el valor **N**, la candidatura del qual estem analitzant. Així que de bon començament formarem la llista **3*N** amb aquests 84 tercets i, amb cada candidat, **TERCETS** únicament en prendrà els que tinguin aquest valor. Atenció, perquè si ens podem permetre aquesta simplificació és gràcies a que els treballem com a tercets implícits (els explícits hi van apareixent com a casos particulars): fixeu-vos que en les comprovacions **A**, **B** i **C** sempre comencem indagant si el tercet és (~~123~~) o (123); suposant que **N** = **2**, caldria usar les 28 plantilles 1-2-3, 1-2-4, ... 2-8-9. Una cosa ben diferent seria una cerca partint de la detecció de tercets explícits, però ja hem vist que aquesta via no cobreix totes les possibilitats.

D'altra banda, no podem ignorar que el conflicte entre tercets implícits d'alguna d'aquestes configuracions i alineacions ortogonals pot esclatar directament per la presència de **N** en totes dues localitzacions (completant l'alineació fins a dur-la a un estat crític, si tenim una **AAA-AAA** o **AAA-BBB** que inclogui el valor **N**, o bé un tercet explícit **AAA-A<u>A</u>A** o **AAA-B<u>B</u>B**, si l'alineació ja té una **N**), amb independència que **ACTUALITZA-PLUS** pugui irrompre en escena emplenant forats a conseqüència de la presència esmentada, o bé que siguin els emplenaments automàtics derivats de **N** els qui acabin de madurar la situació, deixant-la vista per a una sentència negativa.

La depuració definitiva dels elements de **LLL** (això que ens ha donat per anomenar explicitació de tercets implícits) només s'esdevé quan el candidat **N** que examinem és **<u>N</u>**, la baula perduda que representem amb una **<u>A</u>** o **<u>B</u>** en parlar de configuracions **AAA-A<u>A</u>A** i **AAA-B<u>B</u>B**. En relació a l'últim paràgraf, pot respondre a dues situacions:
1) Que aquesta baula la introdueixi directament l'usuari.
2) Que **ACTUALITZA-PLUS** l'hi posi automàticament, a partir d'un emplenament amb **M** (pot ser **M≠N**) efectuat per l'usuari en una altra casella.

Res d'això no pot passar si hi ha una alineació ortogonal (fila o columna) amb el nombre suficient de valors del tercet completat per **<u>N</u>** (3, quan aquest només surt en un requadre 9×3 o 3×9; 2, quan ho fa en dos) com per entrar-hi en contradicció i impedir-ho, però el veredicte negatiu del candidat **<u>N</u>** serà ràpid: en **AAA-B<u>B</u>B** ho serà perquè **TERCETS** (i **FILS/COLS**, si s'escau) hi intervindrà depurant amb caràcter

provisional els elements de **R-LLL**, cosa que provocaria que, com passa en **AAA-AAA**, s'infringís la **norma 2**/**3** i això fos detectat, ... si no fos perquè el dispositiu detector **EXPL-SUBCONJ** és anterior a **TERCETS** en el processat de cada requadre 3×3 (ambdós se situen en **EXPLORA-3*9**), com veurem de seguida, però això no obstarà perquè a la següent autorecursió de **RE-MEMBER** (primera amb **INI** = **nil**) l'exploració sigui interrompuda en descobrir-se que hi ha una casella lliure i sense candidats (per evitar la llarga notació **'(() () () () () () () () ())**, en la pròxima versió de **RE-MEMBER** ho representarem com a **(car NIL*NIL)**, havent definit **NIL*NIL** com una llista amb 9 elements d'aquests, que va bé per inicialitzar la pseudomatriu **LLL**).

Per contra, quan el candidat **N** que examinem completa una alineació contradictòria, això vol dir que ja hi havia configuracions **AAA-AAA** o **AAA-BBB** consolidades, i el veredicte negatiu també serà ràpid. Ara bé, ¿què passarà si la configuració encara no estava consolidada però algun dels emplenaments automàtics desencadenats pel candidat **N** fa de baula perduda recuperada?: doncs el mateix que comentàvem abans, amb l'alineació ortogonal preexistent. Els dubtes poden venir per una altra banda, i aquesta és la raó que haguem efectuat aquest incís, just després d'aclarir que **TERCETS** es limitarà a verificar la presència dels 28 tercets de **3*N** que incloguin el candidat **N**: tenint present que **N** pot ser la baula que completi una configuració i també, mitjançant **ACTUALITZA-PLUS**, una alineació ortogonal amb un valor **M≠N**, o pot ser la casella que completi una alineació ortogonal i també (**ACTUALITZA-PLUS**) una configuració amb un valor **M≠N**, ¿no hauríem de prendre, alternativament, les 28 que incloguessin l'últim valor **M** assignat automàticament per **ACTUALITZA-PLUS**? No caldrà perquè, quan **N** completi un tercet i **M** una alineació ortogonal [o quan **N** completi una alineació ortogonal i **M** un tercet]:
- Si **M** [**N**] participa del mateix tercet que **N** [**M**] no hi haurà cap problema, perquè el tercet figurarà entre els 28 que hem triat.
- Si **M** [**N**] no hi participa:
  - Si la configuració és **AAA-AAA** no hi haurà conflicte amb l'alineació ortogonal.
  - Si la configuració és **AAA- BBB** i hi ha conflicte entre el tercet amb **N** [**M**] i l'alineació ortogonal amb **M** [**N**] no importa, perquè **M** [**N**] participarà d'algun dels dos altres tercets de la configuració, tercet que també serà explicitat i el veredicte negatiu de **N** [**N**] serà ràpid, en veure que la intersecció d'aquest tercet depurat amb l'alineació ortogonal és una casella buida sense candidats.

Línies enrera hem avançat que s'accedeix a **TERCETS** des d'**EXPLORA-3*9** per aprofitar la seva infraestructura de cerca per requadres 3×3. Ara afegirem que l'accés es fa un cop comprovat que no hi ha infracció a la **norma 1** ni a les **normes 2**/**3**, decisió que respon a la precaució elemental de no embrancar-se en un procés laboriós sense abans haver-se assegurat això; no fos cas que el temps esmerçat per **TERCETS** hagués estat inútil. I, com que finalment la verificació de compliment de les **normes 2**/**3** ha quedat confinada al moment **INI** = **T** de **RE-MEMBER**, el detector de configuracions únicament intervindrà en la recursió **INI**cial d'aquesta funció, cosa lògica d'altra banda, si tan feixuga suposem que ha de ser la seva presència. Com que **ACTUALITZA** precedeix **EXPLORA-3*9**, també el dispositiu tapaforats **ACTUALITZA-PLUS** < **ACTUALITZA** haurà actuat abans que **TERCETS**. Acabem de veure que aquesta actuació mai no podrà ser la que provoqui un conflicte relacionat amb configuracions **AAA-AAA** o **AAA-BBB**, perquè en el pitjor dels casos no hi interferirà i en el millor podrà assolir un veredicte negatiu per a **N** sense que **TERCETS** intervingui (quan hi havia intervingut prèviament per explicitar tercets implícits on hi participava un valor **M** utilitzat després per tapar un forat). I sempre és millor ajornar la intervenció de **TERCETS** amb un candidat **N**, perque així ens l'estalviarem en el cas de veredicte negatiu i també en el cas d'un veredicte positiu que no es vegi ratificat per l'elecció de l'usuari. De tota manera, això és una simple valoració *a posteriori* perquè, tal i com ho tenim muntat, ja hem dit que la decisió d'integrar **TERCETS** en **EXPLORA-3*9** comporta la prèvia intervenció d'**ACTUALITZA-PLUS**. Tanmateix, encara queden opcions sobre les quals pronunciar-se i una d'elles, fins ara marginada i que té a veure amb la innecessària repetició d'**ACTUALIZA**, evocada fa un moment, l'exercirem ara: recordem que aquesta funció (amb les seves incursions a **ACTUALITZA-PLUS**) s'executa prospectivament des de **RE-MEMBER** per a cadascun dels candidats i, després d'això, l'execució corresponent a la fase **INI**cial es repeteix des d'**OMPLE-SUDOKU** amb el candidat validat que escull l'usuari, però ara seria poc justificable repetir el tàndem **ACTUALITZA** + **TERCETS**, amb l'agreujant que, en absència **d'EXPLORA-3*9** (que òbviament no cal repetir, perquè ratifiquem un candidat que ja ha provat respectar les **normes 1** i **2**/**3**), a la segona d'aquestes funcions caldrà muntar-li expressament l'organització de la visita per requadres 3×3, almenys quan l'anterior execució de **TERCETS** amb el candidat hagi revelat que hi havia tercets implícits a explicitar.

Evitar reiteracions només serà possible guardant informació obtinguda en la fase **INI**cial (**INI** = **T**) de l'actuació prospectiva de **RE-MEMBER** < **MASCARA** (**REAL** = **nil**) i recuperant-la en l'accés definitiu a **ACTUALITZA**, que efectuem des d'**OMPLE-SUDOKU** un cop escollit candidat per a la casella actual (**REAL** = **T**). Aquesta informació es refereix a valors **A-9*9** pel que fa als forats que emplena **ACTUALITZA-PLUS** i **A-LLL** pel que fa als tercets implícits explicitats per **TERCETS** i **FILS/COLS**, i recollir-la i recuperar-la presenta dues dificultats addicionals: **OMPLE-SUDOKU** actualitza les pseudomatrius 9×9 originals ****9*9**** i **LLL**, mentre que **RE-MEMBER** treballa amb les transformades **0-**9*9**** i **0-LLL**, assignades a **R-9*9** i **R-LLL** respectivament; **TERCETS** (aplicació dels postulats **A** i **B**) necessita processar **R-LLL** reorganitzat en requadres 3×3, però **FILS/COLS** (postulat **C**) torna a actuar per files. En síntesi, ens en sortirem diferenciant la missió d'**ACTUALITZA** segons estigui o no activada **REAL** en el moment d'accedir-hi, guardant a la pseudomatriu 9×9 **REPLUS** els forats emplenats per **ACTUALITZA-PLUS** quan en la casella **P** provem un candidat **N** (**INI** = **T**) i sobretot creant les llistes **N/R-L** i **TN/R-L**, que a diferència de **REPLUS** (local a **RE-MEMBER**) són dues variables locals a **OMPLE-SUDOKU** perquè emmagatzemen informació relativa a tots els candidats **N** (per tal que **ACTUALITZA** la pugui usar sense perdre el temps repetint en situació **REAL** = **T** les operacions que ja s'havien executat en situació **REAL** = **nil**): ambdues són llistes d'associacions, creades mitjançant la funció **3*3/R-L**, en què l'element clau de cada membre és **N** i el segon, al seu torn, una subllista d'associacions amb les coordenades de la casella inferior-esquerra dels requadres 3×3 com a clau de cada membre i el fragment de **R-LLL** corresponent a cada requadre com a contingut referenciat; així com **N/R-L** inclou una llista per cada candidat **N** i aquesta inclou els nou requadres 3×3 de **R-LLL**, **TN/R-L** únicament es crea si **TERCETS** detecta tercets implícits, només amb els membres corresponents als candidats **N** en què es dóna aquesta circumstància i només amb els requadres on n'hagi detectat, i la informació relativa als valors admissibles en cada casella ja la guarda depurada de brossa (és a dir, amb els tercets explicitats pel que fa als postulats **A** i **B**). Però, una vegada la funció **RE-ACT-LLL** ha reconstruït **R-LLL** a partir dels requadres 3×3 de **N/R-L** o **TN/R-L**, i **FILS/COLS** l'ha examinada per files per si fóra d'aplicació el postulat **C**, l'eruga **TN/R-L** encara fruirà d'una segona vida com a papallona, metamorfosant el contingut referenciat dels membres on **A**, **B** o **C** hagin deixat petjada en una llista composta de dos elements: **R-LLL** i **REPLUS**, tots dos reestructurats per **REORD** per tal de restituir-los l'ordenació original. Aquesta nova **TN/R-L** ja estarà en disposició de transferir a **ACTUALITZA** tota la informació que la funció necessita per actualitzar **A-9*9** i **A-LLL** (****9*9**** i **LLL**) quan és reclamada des d'**OMPLE-SUDOKU** després que l'usuari hagi optat entre els candidats **N** compatibles. Si el lector repassa el codi que oferirem tot seguit, potser l'estranyarà que l'esmentat desdoblament d'**ACTUALITZA** vingui en la forma

```
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn ...<1>...)
      (while (not PLUS) ...<2>...))
```

en comptes de limitar-lo a

```
  (if REAL
      (progn ...<1>...)
      (while (not PLUS) ...<2>...))
```

que és el que havíem anunciat. La diferència no vindrà pas pel fet que la variable **TT**, també utilitzada a la fase **INI**cial de **RE-MEMBER** amb altres propòsits, provoqui alguna fuita no prevista des del conjunt d'operacions **<2>** cap al conjunt **<1>**, sinó a l'inrevés: que, quan sigui **(and REAL (not (setq TT (cdr (assoc A-N TN/R-L)))))**, s'executarà el conjunt d'operacions **<2>**. Això vol dir que, si **TERCETS** no ha trobat tercets implícits per explicitar, seguirem com en les versions precedents i, quan l'usuari decideixi amb quin valor emplena la casella actual, **ACTUALITZA** repetirà la detecció i obturació dels forats oberts. Per evitar-ho, aconseguint que totes les actualitzacions virtuals anessin per una banda i totes les reals per un altra, només hauria calgut fer **(REORD REPLUS)** i emmagatzemar la informació relativa als forats a **TN/R-L** fins i tot quan no hi haguessin configuracions **AAA-AAA** ni **AAA-BBB** (de tota manera, **REPLUS** es va actualitzant des de **PLUS-N->P** < **ACTUALITZA-PLUS**), però hem cregut que aquesta reconducció dels casos trivials encara portaria més enrenou, potser erròniament en el cas de no haver-hi configuracions crítiques però sí forats per tapar, però de segur que no en el cas de no haver-hi tampoc forats. A més, recordeu allò que havíem subratllat dues pàgines enrera: els únics casos en què la feina de **TERCETS**, explicitant tercets implícits, podrà ser aprofitada serà, òbviament, quan el veredicte sobre la compatibilitat de **N** sigui positiu, en ser **N** la baula que faltava per completar els tercets explícits de configuracions **AAA-AAA** o **AAA-BBB**, i el requadre (9×3 o 3×9) corresponent no entri en conflicte amb cap alineació ortogonal; tots els altres casos d'explicitació de tercets implícits són per evitar que el veredicte negatiu sobre un candidat **N** no ens faci esperar massa.

Com que probablement el llarg paràgraf precedent queda massa espès com per passar directament al codi sense solució de continuïtat, tractarem de fer una descripció seqüencial, distingint l'etapa prospectiva (**REAL** = **nil**) de l'executiva (**REAL** = **T**). Prèviament, cal haver pres nota de les transformacions aplicades a ****9*9**** i **LLL** per obtenir **0-**9*9**** i **0-LLL**, que **RE-MEMBER** anomena **R-9*9** i **R-LLL** respectivament.

A la fase **INI**cial de **RE-MEMBER**, assignem virtualment a la casella **P** el candidat **N**:
- Els forats emplenats per **ACTUALITZA-PLUS** són emmagatzemats a **REPLUS**.
- Mentre **(not FORA)**, **EXPLORA-3*9** va explorant els requadres 3×3. En cada requadre:
  - Si s'infringeixen la **normes 1** o **2**/**3**, s'activa **FORA**.
  - Si no, s'executa **TERCETS**, que:
    - Copia la informació **LLL** del tercet en la llista de **N/R-L** corresponent a **N**.
    - Mentre **(not TERC)**, compara cadascuna de les 28 tríades de valors 1... 9 que inclouen **N** amb els tercets de caselles horitzontals i verticals, per veure si són aplicables els postulats **A** i/o **B**. Si ho són:
      - Efectua la depuració que pertoca (si era **A**, passarà a considerar **B**).
      - Copia la informació depurada en la llista de **TN/R-L** corresponent a **N**.
      - Activa **TERC** (si això ha passat examinant files, ja no examinarà columnes).
- Posa **INI** al valor de **FORA**. Si **(not INI)**:
  - Si hi ha hagut explicitació de tercets implícits (en aquest cas, a **TN/R-L** hi haurà un membre associat al valor **N**):
    - Reconstrueix **R-LLL** amb la informació **TN/R-L** dels requadres 3×3 modificats i amb la informació **N/R-L** dels no modificats, mijançant **RE-ACT-LLL**.
    - **FILS/COLS** explora files i columnes d'aquest **R-LLL**, per veure si és aplicable el postulat **C** i, si s'escau, les depura.
    - Substitueix per **R-LLL** i **REPLUS** (**REORD** desfarà els canvis que van convertir ****9*9**** i **LLL** en **0-**9*9**** i **0-LLL**) la informació indexada per **N** en **TN/R-L**.

En **ACTUALITZA** < **OMPLE-SUDOKU**, un cop assignat a **P** un dels valors **N** acceptats:
- Si a la llista de associacions **TN/R-L** creada en la primera etapa hi ha un membre associat al valor **N** escollit, es recupera el conjunt d'emplenaments automàtics **REPLUS** que l'assignació de **N** a **P** havia desencadenat (recollint-lo a ****9*9**** i situant-lo definitivament a l'escaquer) i un **LLL** no només actualitzat als nous ocupants sinó que inclou la depuració de tercets implícits efectuada per **TERCETS** i **FILS/COLS**, perquè ambdós paquets d'informació hi van ser emmagatzemats.
- Si no, **ACTUALITZA-PLUS** haurà de repetir la localització de forats a emplenar.

Mostrarem els canvis començant per **INI-LLL**, funció que ara disposa d'un argument per poder-la aprofitar en dues ocasions (just en acabar aquests fragments de codi, veureu aquests dos accessos en les primeres línies d'**OMPLE-SUDOKU**).

```
(defun INI-LLL (L)
  (setq LL () LLL ())
  (repeat 9 (setq LL (cons L LL)))
  (repeat 9 (setq LLL (cons LL LLL))))

(defun ACT-9*9 (N P L1 / L2)
  (reverse (foreach F L1 (setq L2 (cons (if (member P F) (subst N P F) F) L2)))))

(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))
      (if INI (setq REPLUS (ACT-9*9 N K REPLUS)))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L PLUS)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM" "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
```

```
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P)
                   A-9*9 (ACT-9*9 A-N A-P A-9*9)
                   A-LLL (ACT-LLL X Y A-LLL))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))
```

A **EXPLORA-3*9** sols ha variat la penúltima línia (i la precedent, és clar), però no seria gens aclaridor limitar-se a reproduir les noves funcions derivades, **3*3/R-L** i **TERCETS** (a més de **K2-** i **TRANSP-3*3**, derivades d'aquesta) sense veure'n l'accés.

```
(defun 3*3/R-L (/R-L)
  (setq I (list (list X Y) R-L)
        /R-L (if /R-L
                 (if (setq J (assoc N /R-L))
                     (subst (cons N (cons I (cdr J))) J /R-L)
                     (cons (list N I) /R-L))
                 (list (list N I)))))

(defun K2- (K)
  (append (reverse (cdr (member K (reverse K2)))) (cdr (member K K2))))

(defun TRANSP-3*3 (L / X Y E EE EEE) ; Més simple que TRANSPOSAR.
  (setq X 3)
  (repeat 3
    (setq X (1- X) Y 3 EE ())
    (repeat 3 (setq E (nth X (nth (setq Y (1- Y)) L)) EE (cons E EE)))
    (setq EEE (cons EE EEE))))

(defun TERCETS (<R-9 <R-L / R-9 R-L TRANSP TERC SEGUIR I J K K1 K2)
  (setq R-9 <R-9 R-L <R-L N/R-L (3*3/R-L N/R-L))
  (foreach 3N 3*N
    (if (and (not TERC) (member N 3N))
        (progn
          (setq R-9 <R-9 R-L <R-L
                N1 (car 3N) N2 (cadr 3N) N3 (last 3N)
                TRANSP ())
          (repeat 2
            (if (not TERC)
                (progn
                  (setq I () J -1 K 0)
                  (mapcar '(lambda (F/R-9 F/R-L)
                             (setq J (1+ J) K1 0 K2 0)
                             (mapcar '(lambda (E/R-9 E/R-L)
                                        (if (and (or E/R-L (= N1 E/R-9)
                                                     (= N2 E/R-9) (= N3 E/R-9))
                                                 (or (not E/R-L) (member N1 E/R-L)
                                                     (member N2 E/R-L)
                                                     (member N3 E/R-L)))
                                            (setq K1 (1+ K1))
                                            (setq K2 (1+ K2))))
                                      F/R-9 F/R-L)
                             (if (= K1 3) (setq I J))
                             (if (= K2 3) (setq K (1+ K))))
                          R-9 R-L)
                  (if (and I (= K 2) (setq SEGUIR T TERC T))
                      (progn
                        (mapcar '(lambda (E/R-9 E/R-L)
                                   (if TERC
                                       (foreach E E/R-L
                                         (if (and TERC E (not (member E 3N)))
                                             (setq TERC ())))))
                                (nth I R-9) (nth I R-L))
                        (if (not TERC)
                            (progn
                              (setq TERC (nth I R-L) J ())
```

```
                                (foreach L TERC
                                  (setq J (cons (if L
                                                    (progn
                                                      (setq K ())
                                                      (foreach E L
                                                        (setq K
                                                              (cons
                                                                (if (member E 3N) E)
                                                                K)))
                                                      (setq K (reverse K))))
                                                J)))
                                (setq J (reverse J) R-L (subst J TERC R-L))))
                          (setq J -1 K1 ())
                          (foreach N (reverse L1A9)
                            (if (not (member N 3N)) (setq K1 (cons N K1))))
                          (while (and SEGUIR (< J 3))
                            (setq K2 K1 J (1+ J))
                            (if (/= J I)
                                (progn
                                  (mapcar '(lambda (E/R-9 E/R-L)
                                             (if E/R-L
                                                 (foreach N E/R-L
                                                   (if (member N K2)
                                                       (setq K2 (K2- N))))
                                                 (setq K2 (K2- E/R-9))))
                                          (nth J R-9) (nth J R-L))
                                  (if (= (length K2) 3)
                                      (progn
                                        (setq SEGUIR () K1 ())
                                        (foreach L (setq K (nth (- 3 (+ I J)) R-L))
                                          (if L
                                              (progn
                                                (setq I () J I)
                                                (foreach E L
                                                  (setq J
                                                        (cons
                                                          (if E
                                                              (if (member E K2)
                                                                  E
                                                                  (not (setq I T))))
                                                          J)))
                                                (if I (setq K1 (subst (reverse J)
                                                                      L K))))))
                                        (if K1 (setq R-L (subst K1 K R-L)
                                                        TERC T)))))))
                    (if TERC (setq R-L (if TRANSP (TRANSP-3*3 R-L T) R-L)
                                   TN/R-L (3*3/R-L TN/R-L)))))
              (if (not (or TRANSP TERC))
                  (setq R-9 (TRANSP-3*3 R-9 ())
                        R-L (TRANSP-3*3 R-L T) TRANSP T)))))))))

(defun EXPLORA-3*9 (/ FORA R-9 R-L L LL I M) ; EXPLORACIÓ NOMÉS PER REQUADRES 3*3.
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y (car 2R) (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y (cadr 2R) (list X Y) (list (+ X 2) (+ Y 2))) L ())
            (foreach F R-9
              (foreach E F (if (atom E) (setq L (cons E L)))))
            (foreach F R-L
              (foreach E F
                (if E (foreach M E
                        (if (and M (not (member M L))) (setq L (cons M L)))))))
            (if (< (length L) 9) (setq FORA T))
            (if (and INI (not FORA))
                (progn
                  (setq L ())
```

```
                  (foreach F R-L
                    (foreach E F (if E (progn
                                         (setq I ())
                                         (foreach M E (if M (setq I (cons M I))))
                                         (setq L (cons I L))))))
                  (EXPL-SUBCONJ ())
                  (if (not FORA) (TERCETS R-9 R-L))))))))
  FORA)

(defun APLEGA (P1 P2 P3 / L1 L2 L3)
  (setq L1 (assoc P1 TT) L1 (cadr (if L1 L1 (assoc P1 K)))
        L2 (assoc P2 TT) L2 (cadr (if L2 L2 (assoc P2 K)))
        L3 (assoc P3 TT) L3 (cadr (if L3 L3 (assoc P3 K))))
  (list (append (car  L1) (car  L2) (car  L3))
        (append (cadr L1) (cadr L2) (cadr L3))
        (append (last L1) (last L2) (last L3))))

(defun RE-ACT-LLL (/ K)
  (setq K (cdr (assoc N N/R-L))
        K (append (APLEGA '(0 0) '(3 0) '(6 0))
                  (APLEGA '(0 3) '(3 3) '(6 3))
                  (APLEGA '(0 6) '(3 6) '(6 6)))))

(defun INVERT-JJ (KK / II)
  (foreach E '(2 1 0) (setq II (cons (- 3 (length (member E KK))) II))))

(defun FILS/COLS (/ TRANSP SEGUIR I J K L LL M M1 3/R-9 3/R-L)
  (repeat 2
    (if TRANSP (setq R-9*9 (TRANSPOSAR R-9*9) R-LLL (TRANSPOSAR R-LLL)))
    (setq M1 () M ())
    (mapcar '(lambda (F/R-9 F/R-L)
               (setq K1 () K2 L1A9 K -2)
               (repeat 3
                 (setq K (+ K 3) SEGUIR T LL ()
                       3/R-9 (list (nth (1- K) F/R-9) (nth K F/R-9)
                                   (nth (1+ K) F/R-9))
                       3/R-L (list (nth (1- K) F/R-L) (nth K F/R-L)
                                   (nth (1+ K) F/R-L)))
                 (mapcar '(lambda (E/R-9 E/R-L)
                            (if SEGUIR
                                (progn
                                  (if E/R-L
                                      (progn
                                        (setq L ())
                                        (foreach E E/R-L
                                          (if (and E (not (member E LL)))
                                              (setq L (cons E L))))
                                        (if (> (length L) 3)
                                            (setq SEGUIR ())
                                            (setq LL (append L LL))))
                                      (setq LL (cons E/R-9 LL)))
                                  (if (and SEGUIR (> (length LL) 3))
                                      (setq SEGUIR ())))))
                          3/R-9 3/R-L)
                 (if SEGUIR
                     (progn
                       (foreach N K2 (if (member N LL) (setq K2 (K2- N))))
                       (setq K1 (append K1 (list 3/R-L))))
                     (setq I 3/R-L)))
               (if (and (= (length K1) 2) (= (length K2) 3))
                   (progn
                     (setq J ())
                     (foreach L I
                       (setq J (cons (if L (progn
                                             (setq K ())
                                             (foreach E L
                                               (setq K (cons (if (member E K2) E)
                                                             K)))
```

```
                                          (setq K (reverse K))))
                                    J)))
                        (setq J (reverse J)
                              F/R-L (cond ((equal F/R-L (append I (car K1)
                                                               (cadr K1)))
                                           (append J (car K1) (cadr K1)))
                                          ((equal F/R-L (append (car K1) I
                                                               (cadr K1)))
                                           (append (car K1) J (cadr K1)))
                                          (T
                                           (append (car K1) (cadr K1) J))))))
                 (setq M1 (if M (append M1 (list M)) M F/R-L))
              R-9*9 R-LLL)
    (setq R-LLL (append M1 (list M)) TRANSP (not TRANSP)))
  (setq R-9*9 (TRANSPOSAR R-9*9) R-LLL (TRANSPOSAR R-LLL)))

(defun REORD (K / I J 3J JJ)
  (if INV-F (foreach F (reverse K) (setq J (cons (reverse F) J))) (setq J K))
  (if JJ1
      (progn
        (setq K () I -1)
        (repeat 3
          (setq 3J ())
          (repeat 3
            (setq I (1+ I))
            (setq 3J (cons (nth I J) 3J)))
          (setq 3J (reverse 3J)
                JJ (nth (/ I 3) JJ1)
                JJ (if JJ (INVERT-JJ JJ))
                K (append K (if JJ (F=3 3J ()) 3J)))))
      (setq K J))
  (if JJ2
      (setq JJ (INVERT-JJ JJ2) J (F=3 K T))
      (setq J K))
  (if C->F (TRANSPOSAR J) J))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ())
              INI (EXPLORA-3*9))
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
  (if (not (or INI OK))
      (if (not (setq OK (COMPLET-9*9 R-9*9)))
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (if (equal (cadr 2R) NIL*NIL)
                        (setq OK T)
                        (progn
                          (foreach F (cadr 2R)
                            (if (and (not OK) (member (car NIL*NIL) F))
                                (setq OK T)))
                          (if (not OK) (setq OK (EXPLORA-3*9)))
                          (if OK
                              (setq OK ())
                              (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)
```

MASCARA retorna a la penúltima versió (de primer reordenació dels requadres 9×3 i després reordenació de les seves files). Tot queda igual, tret d'algun ajust degut al senyal **INV-F** i als petis canvis efectuats a **F=3** i a **PERMUT-JJ**.

```
(defun F=3 (9*9 9*3 / 9**9)
  (foreach J JJ
    (if 9*3
        (progn
          (setq K (1- (* 3 J)))
          (repeat 3 (setq K (1+ K) 9**9 (cons (nth K 9*9) 9**9))))
        (setq 9**9 (cons (nth J 9*9) 9**9))))
  (reverse 9**9))

(defun PERMUT-JJ (NUM)
  (setq JJ (if (and (>= (car NUM) (cadr NUM))
                    (>= (cadr NUM) (last NUM)))
               () ; Cas trivial identitat: '(0 1 2)
               (if (and (<= (car NUM) (cadr NUM))
                        (<= (cadr NUM) (last NUM)))
                   '(2 1 0)
                   (if (>= (car NUM) (last NUM))
                       (if (> (cadr NUM) (last NUM))
                           '(1 0 2)
                           '(0 2 1))
                       (if (> (cadr NUM) (last NUM))
                           '(1 2 0)
                           '(2 0 1)))))))

(defun MASCARA (L / INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL 0-L *P P* *PP*)
  (setq INV-F () JJ1 () N/R-L () TN/R-L () LL () SS (ssadd))
  (if POST
      (progn
        (setq N (N-3*3 **9*9**) Ñ (TRANSPOSAR N)
              N (1+2+3 N) Ñ (1+2+3 Ñ))
        (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
            (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
        (PERMUT-JJ N)
        (setq JJ2 JJ)
        (if JJ
            (progn
              (setq 0-LLL () I -1)
              (foreach EE (F=3 **9*9** T)
                (setq 0-L () I (1+ I))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) I)) 0-L)))
                (setq 0-LLL (cons (reverse 0-L) 0-LLL)))
              (setq 0-**9*9** (reverse 0-LLL) 0-LLL (F=3 LLL T)
                    P (list X (+ (rem Y 3)
                                 (* 3 (- 3 (length (member (/ Y 3) JJ))))))))
            (setq 0-**9*9** **9*9** 0-LLL LLL))
        (setq *P () P* () J -1)
        (repeat 3
          (setq 0-9 () 0-L () N ())
          (repeat 3
            (setq J (1+ J) K 0)
            (foreach E (nth J 0-**9*9**)
              (if (or (atom E) (equal E P)) (setq K (1+ K))))
            (setq N (cons K N)
                  0-9 (cons (nth J 0-**9*9**) 0-9) 0-L (cons (nth J 0-LLL) 0-L)))
          (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
          (PERMUT-JJ N)
          (setq JJ1 (cons JJ JJ1))
          (if JJ
              (progn
                (setq *PP* () I (- J 3))
                (if (and (> (cadr P) I) (<= (cadr P) J))
                    (setq P (list X (1+ (- J (length (member (rem (cadr P) 3)
                                                             JJ)))))))
```

```
                (foreach EE (F=3 0-9 ())
                  (setq 0-9 () I (1+ I))
                  (foreach E EE
                    (setq 0-9 (cons (if (atom E) E (list (car E) I)) 0-9)))
                  (setq *PP* (cons (reverse 0-9) *PP*)))
                (setq *P (append *P (reverse *PP*))
                      P* (append P* (F=3 0-L ())))))
              (setq *P (append *P 0-9) P* (append P* 0-L))))
        (setq JJ1 (if (equal JJ1 '(() () ())) () (reverse JJ1))
              0-**9*9** *P 0-LLL P*)
        (if (< (car Ñ) (caddr Ñ))
            (progn
              (setq INV-F T K ())
              (foreach EE (reverse 0-**9*9**)
                (setq 0-L ())
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                  0-L)))
                (setq K (cons 0-L K)))
              (setq 0-**9*9** K K ())
              (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
              (setq 0-LLL K P (list (- 8 X) (cadr P)))))))
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (if POST (TREU-RETOL))
          (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                         ((= N 2)     '(0 0.15))
                         ((= N 3)  '(0.15 0.15))
                         ((= N 4) '(-0.15 0))
                         ((= N 5)     '(0 0))
                         ((= N 6)  '(0.15 0))
                         ((= N 7) '(-0.15 -0.15))
                         ((= N 8)     '(0 -0.15))
                         (   T     '(0.15 -0.15))) (itoa N))
          (command "DESIGNA" (ssadd (entlast) SS) "")
          (setq LL (cons N LL)))
        (if POST (TREU-RETOL))))
  (if POST
      (progn
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))
```

Pel que fa a **OMPLE-SUDOKU**, caldrà ampliar el conjunt de variables locals i l'inici del cos també variarà per efecte del nou disseny de la funció **INI-LLL** (que ara té un argument, com hem mostrat abans, encapçalant els nous fragments de codi) i de les variables en funció de constants **NIL*NIL** i **0*0**. De començar així

```
(defun OMPLE-SUDOKU (/ CURSOR P-CURS SS C->F)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq **9*9** (INI-COORDS) P-CURS '(-1.1919 0.4419) POST T)
        (INI-LLL) ...) ...) ...)
```

passa a

```
(defun OMPLE-SUDOKU (/ 0*0 NIL*NIL CURSOR P-CURS C->F INV-F JJ JJ1 JJ2 TT N/R-L
                       TN/R-L)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (INI-LLL ())
        (setq NIL*NIL LLL
              0*0 (INI-COORDS) **9*9** 0*0
              P-CURS '(-1.1919 0.4419) POST T)
       (INI-LLL L1A9) ...) ...) ...)
```

Tres quarts del mateix tindrem a la funció principal, que de començar així:

```
(defun C:SUDOKULUM (/ ANG TG L1A9 CONF ABC G I J K L LL LLL M N N1 N2 N3 N4 P
                      PX PY X Y OK POST OSN FIL SRT SVT ECO **9*9** **9**9**)
  (setq ANG -0.5   ; Un angle que giri el YIN-YANG a una velocitat proporcionada a
        TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9))               ; la del cursor.
  (PREPGRAF-9*9) ...)
```

passa a fer-ho així

```
(defun C:SUDOKULUM (/ ANG TG L1A9 CONF ABC 1-2 3*N G I J K KK KKK L LL LLL M N N1
                      N2 N3 N4 P PX PY Q QX QY SS V X Y OK POST OSN FIL SRT SVT
                      ECO **9*9** **9**9**)
  (setq ANG -0.5   ; Un angle que giri el YIN-YANG a una velocitat proporcionada a
        TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9))               ; la del cursor.
  (foreach N1 L1A9
    (foreach N2 (cdr (member N1 L1A9))
      (foreach N3 (cdr (member N2 L1A9)) (setq 3*N (cons (list N1 N2 N3) 3*N)))))
  (setq 3*N (reverse 3*N))
  (PREPGRAF-9*9) ...)
```

Controlades (de moment) les situacions d'excepcionalitat que les configuracions **AAA-BBB** provocaven en les exploracions autorecursives a càrrec de **RE-MEMBER**, era evident que els SUDOKUs-PROBLEMA deduïbles d'unes SUDOKUs-SOLUCIÓ constituïdes per tres requadres 9×3 o 3×9 d'aquesta mena podien quedar definits amb poc més de 18 caselles plenes. Almenys 18 havien de correspondre als tres parells de tercets que implicaven aquestes configuracions; quantes més en caldrien ja tenia a veure amb l'habilitat per combinar els dos tercets de cada requadre de manera que els valors explícits es condicionessin transversalment, permetent reduir les 20 caselles plenes que veiem en el requadre 9×3 de la dreta (60 en el total de l'escaquer 9×9, on els dos requadres 3×3 amb tercet explícit -en negreta- s'han completat emplenant dues caselles a cadascuna de les altres línies -valors subratllats- i en el tercer requadre això s'ha fet a totes tres) a la meitat més o menys. Per reduir-les a 10 (30 en total) no caldrà que ens hi trenquem el cap: veieu com ens en hem sortit a la solució PANÒNICA de la dreta, procurant no repetir tercets explícits en un mateix requadre 3×9 i limitant l'emplenament a les caselles justes per determinar les permutacions ternàries que feien ballar els implícits. Però si volem seguir aprimant el SUDOKU-PROBLEMA, cal abandonar aquesta tipologia de SUDOKU-SOLUCIÓ, perquè contra totes les aparences hem de seguir procurant que entre els tercets explícits del mateix requadre 9×3 no hi hagi valors repetits però caldrà passar a solucions que, com a la CANÒNICA, la NATURMÍN o a la majoria de les que fins ara han desfilat per aquí, les configuracions **AAA-BBB** a cada requadre es nodreixen de tercets diferents. Així, i passant a la notació habitual més clara (totes les caselles emplenades, en negreta), veurem com amb la PANÒNICA (esquerra) hem d'abandonar l'esquema de tercets explícits per aconseguir baixar el llistó de 30 caselles a 27, mentre que amb la CANÒNICA (centre) aconseguim idèntic resultat (27 caselles plenes) mantenint l'esquema. Haurem de desviar-nos d'aquesta solució (això sí, mantenint l'esquema de tercets explícits) per definir un SUDOKU-PROBLEMA amb només 21 caselles plenes (dreta), llistò que, ara com ara i amb la reserva del que veurem d'aquí a tres capítols (*Condicions mínimes*), sembla un rècord absolut.

7 8 9	1 2 3	4 5 6
4 5 6	**7 8 9**	1 2 3
1 2 3	4 5 6	7 8 9

9 7 8	3 1 2	6 4 5
6 4 5	9 7 8	3 1 2
3 1 2	6 4 5	**9 7 8**

8 9 7	2 3 1	5 6 4
5 6 4	8 9 7	**2 3 1**
2 3 1	**5 6 4**	8 9 7

9 7 **8**	3 1 **2**	6 **4** 5
6 4 **5**	9 **7** 8	3 1 2
3 1 2	6 **4** 5	**9** 7 **8**

8 9 **7**	2 **3** 1	5 **6** 4
5 **6** 4	8 9 7	**2** 3 **1**
2 **3** 1	**5** 6 **4**	8 9 7

7 **8** 9	1 **2** 3	4 5 **6**
4 5 6	**7** 8 **9**	1 **2** 3
1 2 **3**	4 5 6	7 **8** 9

9 1 2	3 4 **5**	6 7 8
6 7 8	9 **1** 2	3 4 5
3 4 5	6 7 8	**9 1 2**

8 9 **1**	**2** 3 **4**	5 6 7
5 6 **7**	8 9 1	**2 3 4**
2 3 4	**5 6 7**	**8** 9 1

8 9 7	1 2 3	**4** 5 6
4 5 6	**7 8 9**	1 2 3
1 2 3	4 5 6	7 8 9

9 **5** 8	**6 4 1**	3 7 2
7 3 2	5 8 9	4 1 6
4 1 6	2 7 **3**	**5 9 8**

6 4 9	1 2 7	8 5 3
3 8 5	9 6 4	**1 2 7**
2 7 1	8 3 5	9 6 4

8 9 7	3 1 2	6 **4** 5
7 9 8	2 3 1	
1 2 3	4 5 6	7 8 9

PANÒNICA (27) CANÒNICA (27) ¿mínima (21)?

Hem tocat aquest tema perquè, justament en un moment de distensió per assaborir la mel de la victòria, quan l'autor creia que la problemàtica de les configuracions **AAA-BBB** ja estava resolta del tot (presumpció certa) i que, en conseqüència, també ho estava la d'exploració de candidats viables (presumpció falsa) i es dedicava a jugar amb tercets explícits per tal de veure fins on aconseguia abaixar el llistó, va produir-se la mateixa situació d'altres vegades (una espera llarga, aquest cop per rebre un veredicte favorable al candidat reticent) quan ja es donava per fet que el muntatge estava preparat per fer front a totes les contingències possibles.

Tot i el desconcert (i la desmoralització: encara no?) per aquest sobtat incident, si l'autor tingués creències religioses (que no en té) hauria d'haver-li agraït al seu àngel de la guarda la mercè d'haver-li permès descobrir el flanc desprotegit, malgrat que el tipus d'experiències que estava realitzant (jugar amb sudokus amb les configuracions esmentades) era favorable a que fallades com la que mostrarem es manifestessin amb menor virulència (és a dir, a que l'espera no fos tan llarga com ho hauria estat en absència de tercets o amb una presència no tractada per les últimes implementacions) i poguessin passar desapercebudes. Però ara convé deixar aquestes consideracions per a després i passar directament al cas que va destapar l'olla. Amb la intenció de veure què passava omplint caselles d'una SUDOKU-SOLUCIÓ afectada per l'alternància dels tercets 1-2-5, **3-6-7** i **4-8-9** en el primer requadre 9×3, els tercets **1-2-3**, 4-6-9 i 7-8-5 (que finalment esdevindria inviable) en el segon i els tercets 1-**2**-6, 3-5-7 i altre cop **4-8-9** en el tercer, ja teníem plenes les que veieu en negreta (esquerra) i havíem activat la 8,8 quan una espera de 6 minuts i quinze segons va acabar amb l'aparició del candidat **1** i, ja sense pauses, dels altres valors viables (3, 4, 5, 6, 7, 8 i 9). Després d'infructuosos intents per ensumar l'origen del mal, vam optar pel conte de la vella i, representant el sudoku tal i com l'havien deixat els mecanismes optimitzadors per reordenació de files i requadres 9×3, sense transposició (centre), vam rastrejar la trajectòria NATURMÍN per tal de veure on havía quedat parat l'avanç per iniciar una regressió, i fins quina posició havia reculat aquesta.

0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	9 5 2	6 8 4	7 3 1
0 0 0	0 0 0	**2 1** 0	0 0 0	0 0 0	0 0 0	6 1 3	7 9 5	8 4 2
8 9 4	0 0 0	0 0 0	0 0 0	**1 2 3**	0 0 0	7 4 8	**1 2 3**	9 5 6
0 0 0	0 0 0	0 0 0	0 0 0	0 0 2	0 0 0	2 7 1	8 3 6	5 9 4
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	**2 1** 0	5 3 6	4 7 9	**2 1** 8
0 0 0	**1 2 3**	0 0 0	**8 9 4**	5 3 1	6 7 X	**8 9 4**	5 1 2	6 7 3
0 0 0	0 0 0	0 0 0	4 8 9	3 6 7	1 2 5	4 8 9	3 6 7	1 2 5
0 0 0	**9 4 8**	0 0 0	1 2 5	**9 4 8**	3 6 7	1 2 5	**9 4 8**	3 6 7
3 6 7	0 0 0	0 0 0	**3 6 7**	2 1 5	4 8 9	**3 6 7**	2 5 1	4 8 9

Com veieu, la primera embranzida ja no ha pogut emplenar la casella 9,4 (X) perquè la presència d'un **2** en el requadre 3×3 centre-dreta impedeix que es completi la 4ª fila: aquí s'inicia una penosa marxa enrera (que ho podría ser encara més, si els últims dispositius incorporats no haguéssin fet explícita a **0-LLL** l'estructura de tres tercets del primer requadre 9×3) que sols s'aturarà quan s'hagin permutat els valors 1 i 5 del segon tercet; a partir d'aquest moment es reprèn la progressió, que sense massa entrebancs arribarà a completar el sudoku (dreta), confirmant la validesa del valor **1** a 8,5 (o 8,8 en l'escaquer 9×3 original, però a partir d'ara ens limitarem a la versió modificada per reordenació dels requadres 9×3).

La primera indagació és localitzar la causa última (no la immediata, que és la presència d'un **2** a 7,5) i esbrinar si és assimilable a alguna de les situacions caracteritzades per infringir allò que dos subcapítols enrera ens va donar (perquè no se'ns va ocórrer res de millor) per anomenar **normes**. En aparença, fins a quedar en punt mort no hi ha cap requadre 3×3 en què es produeixi alguna infracció, però si anem gairebé al principi de l'exploració, amb només dues caselles emplenades virtualment (2 a 4,1 i 1 a 5,1), i aïllem els fragments de **R-LLL** corresponents als requadres esmentats (**R-L** en **EXPLORA-3*9**) ens xocarà el que trobarem en el central:

(0 0 **3 4 5 6 7 8** 0)	(0 0 **3** 0 **5 6 7 8 9**)	(**1 2** 0 **4 5 6 7** 0 **9**)
(0 0 **3 4 5 6 7 8** 0)	(0 0 **3** 0 **5 6 7 8 9**)	(0 0 0 **4 5 6 7** 0 **9**)
(0 0 **3** 0 **5 6 7** 0 0)	(0 0 **3** 0 **5 6 7** 0 0)	(**1 2** 0 0 **5 6 7** 0 0)

No hi ha cap infracció, almenys pel que respecta a aquests requadres (que, en un exercici de realisme era l'únic àmbit que ens vèiem amb cor de supervisar sense que els temps d'espera se'ns disparessin escandalosament). Però sorprèn una cosa: que els valors 1 i 2 només apareguin en les llistes corresponents a les caselles 6,4 i 6,6; és a dir, que només siguin admissibles en aquestes dues posicions del requadre 3×3. De fet, el que després passa a la 4ª fila és conseqüència directa d'això: les tres primeres caselles ja estan ocupades per valors que no són **1** ni **2**; a les tres últimes no admeten cap d'aquests valors, que ja figuren en el requadre centre-dreta, i de les tres caselles intermèdies només n'hi ha una que els admeti. Tan aviat com li assignem un dels dos (seguint la dinàmica NATURMÍN, el valor 1), el forat 6,6 (6ª fila) serà tapat automàticament per l'altre (el 2, assignat per **ACTUALITZA-PLUS**), però la 4ª fila ja estarà condemnada. De manera que la sospita que ens mossegava les entranyes, que no n'hi havia prou a verificar les **normes 1**, **2** i **3** en els requadres 3×3, i que calia fer-ho també per files i per columnes, es confirma i ens cau al damunt com una galleda d'aigua freda, per allò de doblar amb tota seguretat uns temps d'espera que ja pecaven d'excessius. ¡Mireu per on, la denominació **EXPLORA-3*9** (que inicialment havia sorgit d'una triple exploració de subconjunts de nou caselles: files, columnes i requadres 3×3, i que havia quedat inapropiada en limitar-nos a l'última) inesperadament ha recobrat plena vigència! Però no ho hem de pintar tan negre. Efectivament, seria espaordidor que sempre i en qualsevol circumstància haguéssim d'estendre l'acció d'**EXPLORA-3*9** a files i columnes, a més de seguir fent-ho per requadres 3×3, però si recuperem la calma ens adonarem que tampoc no es tracta d'això: la configuració de cada fragment **R-L** ens pot indicar si cal o no ampliar l'exploració. Si prenem com a referència el cas que ha fet saltar l'alarma, no hi ha res de més senzill: només seria qüestió de fer una comprovació suplementària amb els requadres 3×3, inquirint la presència de parells de valors que només apareguin en dues llistes (entenem-nos: que els dos apareguin plegats en dues llistes i que cap d'ells no surti en una altra llista). Tanmateix, no hem de precipitar-nos afirmant que la presència única (en una fila o columna) de dos valors que comparteixen una mateixa llista a **R-L** sols es pot donar en aquestes configuracions, perquè per trobar-ne una altra de conflictiva només cal resituar la casella ocupada (**2**) del requadre 3×3 centre-dreta, tal i com us mostra l'escaquer 9×9 de la dreta. Per cert, que les mateixes configuracions de **R-L** en el requadre central i el mateix conflicte s'esdevindrien si els valors de les caselles 4,1 (**2**) i 5,1 (**1**) ja estiguéssin consolidats (centre, pel que fa al cas de l'última pàgina, i dreta pel que fa a l'actual), amb l'única diferència que el candidat **1** seria inviable), i ens centrarem en aquests dos últims, que tenen l'avantatge de prescindir d'emplenaments virtuals amb l'única excepció de 8,5 (**1**) (4,1 (**2**) i 5,1 (**1**) aquí ja són caselles definitivament ocupades), per ser així de comprensió més immediata en entrar en joc només els elements justos... o gairebé.

0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 0 0	**1 2 3**	0 0 0	0 0 0	**1 2 3**	0 0 0
0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0
0 0 0	0 0 0	0 **1** 0	0 0 0	**2 1** 0	0 **1** 0
8 9 4	5 3 1	6 7 X	**8 9 4**	0 0 0	**8 9 4**
4 8 9	3 6 7	1 5 2	0 0 0	0 0 0	0 0 0
1 2 5	**9 4 8**	3 6 7	0 0 0	**9 4 8**	0 0 0
3 6 7	2 1 5	4 8 9	**3 6 7**	**2 1** 0	**3 6 7**

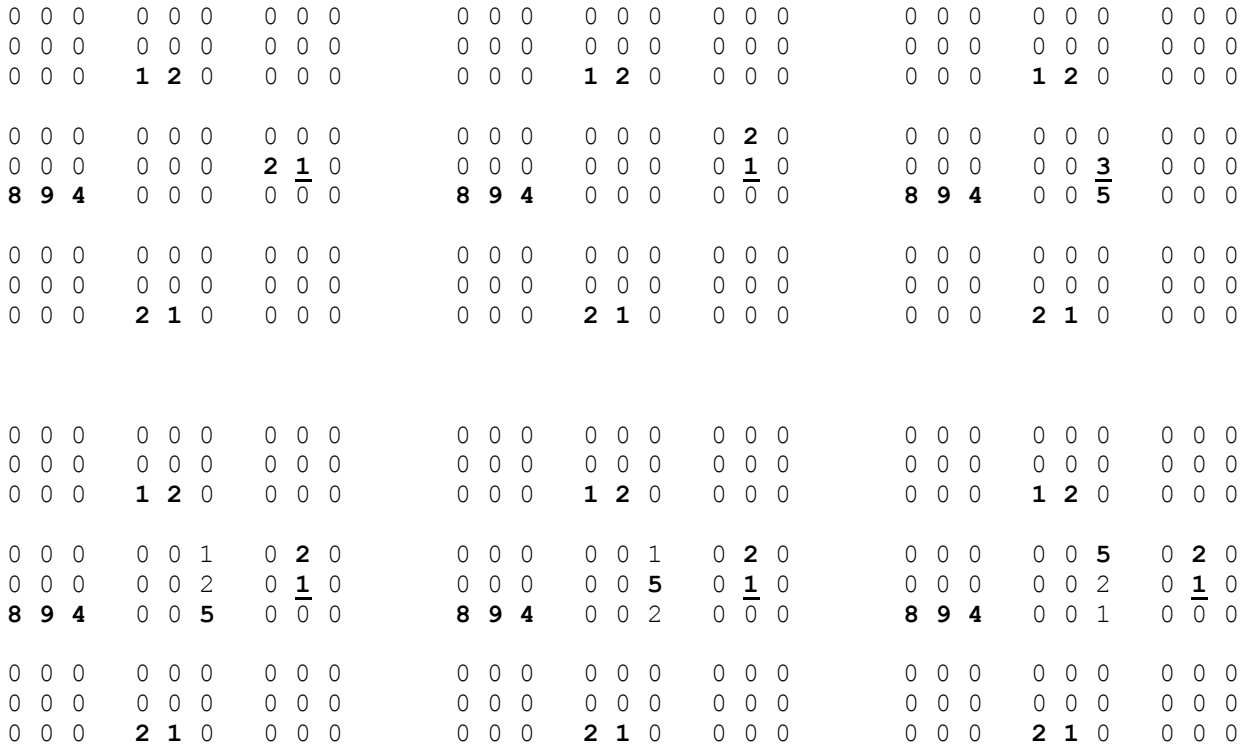
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 **2** 0
0 0 0	0 0 0	0 **1** 0
0 0 0	0 0 0	**8 9 4**
0 0 0	0 0 0	0 0 0
0 0 0	**9 4 8**	0 0 0
0 0 0	**2 1** 0	**3 6 7**

Un altre cop veiem (al mig) que la 4ª línia de **R-L** infringeix les **normes 2**/**3**, però no el fragment corresponent al requadre 3×3 central, que tindrà l'aspecte següent:

(0 0 **3 4 5 6 7 8** 0)	(0 0 **3** 0 **5 6 7 8 9**)	(**1** 0 0 **4 5 6 7** 0 **9**)
(0 0 **3 4 5 6 7 8** 0)	(0 0 **3** 0 **5 6 7 8 9**)	(0 **2** 0 **4 5 6 7** 0 **9**)
(0 0 **3** 0 **5 6 7** 0 0)	(0 0 **3** 0 **5 6 7** 0 0)	(**1 2** 0 0 **5 6 7** 0 0)

Si fa un moment hem emfasitzat allò de "gairebé", era perquè encara no s'ha acabat el bròquil i ens convindria depurar al màxim la presentació dels casos conflictius (depuració en el sentit de reduir a la mínima expressió les caselles plenes, sense que ens hagi de preocupar que les condicions d'emplenament puguin desencadenar la transposició de l'escaquer 9×9 i/o una reordenació per files i tercets). Així que començarem repetint els dos últims casos, desprovistos ara (esquerra i centre a la

primera renglera) de tota la faramalla destinada a reequilibrar les ocupacions per tal que no calgui efectuar transposicions ni permutacions i el sudoku representat coincideixi amb l'utilitzat per **MASCARA**, seguirem amb altres configuracions que, per poc que ens parem a pensar sobre la problemàtica, poden sorgir com a bolets.



Abans que res, pensem en possibles variants del primer cas: si la casella 6,4 o la casella 6,6 estiguessin ocupades per valors diferents de **1** i **2**, les **normes 2**/**3** ja no es complirien a nivell del requadre 3×3, raó per la qual el candidat **1** no seria acceptat, sense necessitat d'arbitrar cap dispositiu per avançar aquest veredicte; pel que fa a la possibitat que ho estiguessin les dues alhora o, en general, dues qualssevol de les caselles 6,4, 6,5 i 6,6, l'hem de descartar fins i tot suposant que en el requadre 3×3 centre-dreta encara no hi ha cap valor **1** ni **2**, ja que per a la segona d'aquestes ocupacions no hi hauria més candidats que **1** i **2** (els altres incomplirien les **normes 2/3**, com podeu deduir fàcilment de la nova representació, la 3ª a la 1ª renglera). Si pensem en variants del segon cas, les coses succeeixen d'una altra manera: si la casella 6,4 estigués ocupada per valors diferents de **1** i **2** (tant hi fa que l'ocupació fos anterior o posterior a la de 8,6 amb el **2**), a 6,5 i 6,6 **ACTUALITZA-PLUS** els assignaria automàticament aquests valors, i el candidat **1** no seria admès per infracció de la **norma 1**, aquest cop a l'àmbit del requadre 3×3 centre-esquerra (absència dels valors **1** i **2** entre **R-9** i **R-L**, com podeu deduir fàcilment de la 1ª representació a la 2ª renglera); si ho estigués la casella 6,5, amb el **2** a 8,6 **ACTUALITZA-PLUS** assignaria automàticament **2** i **1** a 6,4 i 6,6, i el candidat **1** no seria admès per infracció de la **norma 1** a l'àmbit del requadre 3×3 centre-esquerra (absència del valor **1** entre **R-9** i **R-L**, com podeu deduir de la 2ª representació a la 2ª renglera); si ho estigués la casella 6,6, amb el **2** a 8,6 **ACTUALITZA-PLUS** assignaria automàticament **1** i **2** a 6,4 i 6,5, i el candidat **1** no seria admès per infracció de la **norma 1** a l'àmbit del requadre 3×3 centre-esquerra (absència del valor **2** entre **R-9** i **R-L**, com podeu deduir de la 3ª representació a la 2ª renglera); pel que fa a la possibitat que ho estiguessin dues alhora, l'hem de descartar fins i tot abans d'introduir algun valor **1** o **2** en el requadre 3×3 centre-dreta, per les raons adduïdes en el primer cas.

Com tantes altres vegades, com que l'autor frisava per veure aquestes incidències sota control, es va llançar a resoldre-les, en el fons confiant que les hipòtesis contemplades fossin les úniques que poguessin derivar en infracció de les **normes 2/3** a nivell de fila o columna (que en un tercet de requadre 3×3, horitzontal o vertical, els valors atípics -seguirem referint-nos a **1** i **2**- hi figuressin només dos cops, concentrats **1** i **2** en dues llistes o presents **1** i **2** només en una, l'**1** sol en una altra i el **2** sol a la tercera) i pensant que, si malgrat tot sorgien altres casos, s'hi podrien acollir complicant un pèl el dispositiu que tot seguit veureu. La detecció d'aquestes circumstàncies s'inclou en **EXPLORA-3*9** i si se n'han trobat les files/columnes transversals (una o dues) les explora la nova funció **2*FIL/COL:**

```
(defun 2*FIL/COL (FILAS)
  (setq K (car M))
  (repeat I
    (if (not FORA)
        (progn
          (setq L2 (1+ L2) I 0)
          (foreach E (if FILAS
                         (nth (cadr (nth L2 L1)) (cadr 2R))
                         (nth (car  (nth L2 L1)) (TRANSPOSAR (cadr 2R))))
            (if (or (member J E) (member K E)) (setq I (1+ I))))
          (if (= I 1) (setq FORA T))))))

(defun TERCETS (<R-9 <R-L / R-9 R-L TRANSP TERC SEGUIR I J K K1 K2)
  (foreach 3N 3*N
    (if (and (not TERC) (member N 3N))
        (progn
          (setq R-9 <R-9 R-L <R-L
                N1 (car 3N) N2 (cadr 3N) N3 (last 3N)
                TRANSP ())
          (repeat 2
            (if (not TERC)
                (progn
                  ........................
                  (if (not (or TRANSP TERC))
                      (setq R-9 (TRANSP-3*3 R-9 ())
                            R-L (TRANSP-3*3 R-L T) TRANSP T)))))))))

(defun EXPLORA-3*9 (/ FORA R-9 R-L L L1 L2 I J K M); EXPLORACIÓ SOLS PER REQUADRES
  (foreach Y '(0 3 6)                               ; 3*3, PERÒ AMB REPÀS DE DUES
    (foreach X '(0 3 6)                             ; FILES O COLUMNES.
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y (car 2R) (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y (cadr 2R) (list X Y) (list (+ X 2) (+ Y 2))) L ())
            (foreach F R-9
              (foreach E F (if (atom E) (setq L (cons E L)))))
            (if (< (length L) 9)
                (progn
                  (setq L1 ())
                  (foreach F R-L
                    (foreach E F
                      (if E (foreach M E
                              (if (and M (not (member M L)))
                                  (setq L (cons M L) L1 (cons M L1)))))))
                  (if (< (length L) 9)
                      (setq FORA T)
                      (progn
                        (setq L ())
                        (foreach N L1
                          (setq K (1- Y) L2 ())
                          (foreach F R-L
                            (setq K (1+ K) J (1- X))
                            (foreach E F
                              (setq J (1+ J))
                              (if (member N E) (setq L2 (cons (list J K) L2)))))
                          (if (= (length L2) 2) (setq L (cons (cons N L2) L))))
                        (if L ; Ha de ser (> (length L) 1)
                            (while (and (not FORA)
                                        (setq J (caar L) L1 (cdar L) L (cdr L)))
                              (foreach M L
                                (if (not FORA)
                                    (if (or (and (equal (cdr M) L1)
                                                 (setq I 2 L2 -1))
                                            (and (equal (cadr M) (car  L1))
                                                 (setq I 1 L2 -1))
                                            (and (equal (caddr M) (cadr L1))
                                                 (setq I 1 L2 0)))
```

```
                                                (2*FIL/COL (= (caar L1)
                                                              (caadr L1)))))))))))
                    (if (and INI (not FORA))
                        (progn
                          (setq L ())
                          (foreach F R-L
                            (foreach E F
                              (if E (progn
                                      (setq I ())
                                      (foreach M E (if M (setq I (cons M I))))
                                      (setq L (cons I L))))))
                          (EXPL-SUBCONJ ())
                          (if (not FORA) (TERCETS R-9 R-L))))))
                (if (and INI (not FORA)) (setq N/R-L (3*3/R-L N/R-L)))))))
  FORA)
```

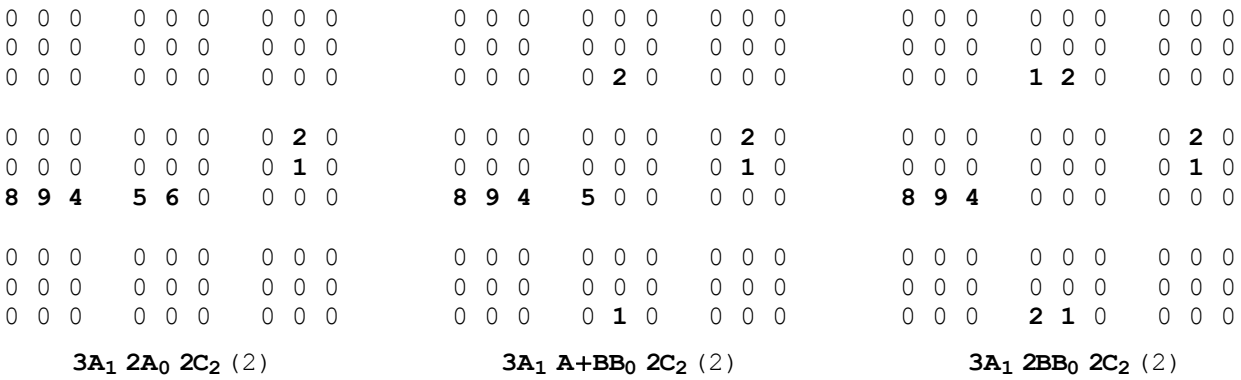
Fixeu-vos que, en complicar-se l'exploració de requadres 3×3 i mirar de compensar-ho afinant la selecció, ens hem adonat que fins ara els que tenien les 9 caselles ocupades passaven innecessàriament per **EXPL-SUBCONJ** i per **TERCETS** (potser tampoc era tan greu perquè a mesura que l'escaquer 9×9 es va poblant disminueix el nombre d'autorecursions **RE-MEMBER**, però no deixava de ser un descuit important). Ara s'hi ha posat remei, però la possibilitat que aquests requadres 3×3 plens puguin formar part, en companyia d'altres més buits, de requadres 9×3 o 3×9 amb una configuració **AAA-AAA** o **AAA-BBB**, ha provocat que eliminem **(setq R-9 N/R-L (3*3/R-L N/R-L))** de la primera línea del cos de la definició de **TERCETS** (per això aquesta definició l'hem reproduïda de forma fragmentària: per apreciar la supressió) i passem l'assignació a la penúltima d'**EXPLORA-3*9**.

Ja hem dit que el codi que acabem de presentar tenia força possibilitats de no ser el definitiu però que, abans de llançar-nos a una anàlisi més exhaustiva, l'autor volia comprovar si aquest dispositiu no disparava excessivament el temps d'espera. Així doncs hem de sistematitzar la recerca, i amb aquest propòsit tractarem d'anar a les arrels del problema. Perquè en tota la 4ª línia no hi hagi cap altre casella que 6,4 amb capacitat per acceptar els valors **1** o **2**, ultra la impossibilitat que una casella d'aquesta línia, de la 6ª columna o del requadre 3×3 central estigui ocupada amb algun dels dos valors, cal assegurar que tampoc siguin admissibles en:

- Les altres dues caselles que té la línia en el requadre central, i per a cada casella això ho podem aconseguir:
  A) Ocupant la casella amb valors **3... 9**.
  BB) Amb la casella lliure, però amb un **1** i un **2** a la columna (fora del requadre central, és clar). Amb la casella plena l'estudiarem després com una variant.
  Si caracteritzem el requadre 3×3 central amb el subíndex **0**, les combinacions possibles són: $\mathbf{2A_0}$, $\mathbf{A{+}BB_0}$ i $\mathbf{2BB_0}$.
- Les tres caselles que té la línia a cadascun dels altres dos requadres, i per a cada casella això ho podem aconseguir:
  A) Ocupant la casella amb valors **3... 9**.
  BB) Amb la casella lliure, però amb un **1** i un **2** a la columna (fora del requadre central, és clar). Amb la casella plena l'estudiarem després com una variant.
  I, per al conjunt de les tres caselles:
  C) Ocupant alguna de les altres 6 caselles del requadre amb el valor **1** o **2**, cosa que fem per a ambdós valors (**2C**) o que haurem de complementar amb algun dels dispositius **A** o **B**, entès aquest segon com aclarim tot seguit.
  B) Amb la casella lliure, però amb un **1** o un **2** a la columna (fora del requadre, és clar). Amb la casella plena l'estudiarem després com una variant.
  Si caracteritzem aquests requadres 3×3 amb els subíndexs **i = 1, 2**, hi ha les combinacions següents: $\mathbf{3A_i}$, $\mathbf{2A{+}BB_i}$, $\mathbf{A{+}2BB_i}$, $\mathbf{A{+}2B{+}C_i}$ i $\mathbf{2C_i}$. Només desdoblarem els casos si les variants d'ubicació de les caselles ocupades influeixen en la configuració de **R-L** en el requadre central (en $\mathbf{A{+}2B{+}C_i}$, per exemple, que **A** i **C** se situïn o no a la mateixa columna no repercuteix en el requadre 3×3 central, raó per la qual només mostrarem i considerarem una de les dues possibilitats).

De les 45 combinacions que resten en eliminar les 30 que són simètriques respecte al requadre 3×3 central, en podríem suprimir d'entrada unes quantes, per òbviament innòcues (com les tres $\mathbf{3C_1\ 2A_0\ 3C_2}$, $\mathbf{3C_1\ A{+}BB_0\ 3C_2}$ i $\mathbf{3C_1\ 2BB_0\ 3C_2}$ en què s'infringeixen les **normes 2/3** a nivell de requadre) o impossibles (com la primera, $\mathbf{3A_1\ 2A_0\ 3A_2}$), a més de les dues presentades abans i que, en la sistematització que ara aplicarem, veiem que responen a una mateixa combinació $\mathbf{3A_1\ 2BB_0\ 2C_2}$, confirmant la necessitat d'estudiar variants amb una repercusió significativa a **R-L** (requadre 3×3 central):

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **5** ? 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **7 3 6** 0 0 0 0 0 0 0 0 0
$3A_1$ $2A_0$ $3A_2$		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **5** 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **7 3 6** 0 0 0 0 0 0 0 0 0
$3A_1$ $A+BB_0$ $3A_2$		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **5** 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **7 3 6** 0 0 0 0 0 0 0 0 0
$3A_1$ $2BB_0$ $3A_2$		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **5** 0 0 0 0 0 0 0 0 0 0 0	0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 **7 3** 0 0 0 **2** 0 0 0 0 0 0
$3A_1$ $2A_0$ $2A+BB_2$		
0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 **2** 0 0 0 0 0 0	0 0 **1** 0 0 0 0 **2** 0 0 0 0 0 0 0 **5** 0 0 0 **1** 0 0 0 0 0 0 0	0 0 **1** 0 0 0 **1 2** 0 0 0 0 0 0 0 **7** 0 0 0 0 **2** 0 0 0 0 0 0
$3A_1$ $A+BB_0$ $2A+BB_2$		
0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 **2** 0 0 0 0 0 0	0 0 **1** 0 0 0 0 **2** 0 0 0 0 0 0 0 **5** 0 0 0 **1** 0 0 0 0 0 0 0	0 0 **1** 0 0 0 **1 2** 0 0 0 0 0 0 0 **7** 0 0 0 0 **2** 0 0 0 0 0 0
$3A_1$ $2BB_0$ $2A+BB_2$		
0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 **2** 0 0 0 0 0 0	0 **2 1** 0 0 0 0 0 0 0 0 0 0 0 0 **5** 0 0 0 **1 2** 0 0 0 0 0 0	0 **2 1** 0 0 0 **1 2** 0 0 0 0 0 0 0 **7** 0 0 0 **1 2** 0 0 0 0 0 0
$3A_1$ $2A_0$ $A+2BB_2$		
0 **2 1** 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 **1 2** 0 0 0 0 0 0	0 **2 1** 0 0 0 0 **2** 0 0 0 0 0 0 0 **5** 0 0 0 **1 2** 0 0 0 0 0 0	0 **2 1** 0 0 0 **1 2** 0 0 0 0 0 0 0 **7** 0 0 0 **1 2** 0 0 0 0 0 0
$3A_1$ $A+BB_0$ $A+2BB_2$		
0 **2 1** 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 **1 2** 0 0 0 0 0 0	0 **2 1** 0 0 0 0 **2** 0 0 0 0 0 0 0 **5** 0 0 0 **1 2** 0 0 0 0 0 0	0 **2 1** 0 0 0 **1 2** 0 0 0 0 0 0 0 **7** 0 0 0 **1 2** 0 0 0 0 0 0
$3A_1$ $2BB_0$ $A+2BB_2$		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 **2** 0 0 0 0 0 0	0 0 0 0 0 0 0 **2** 0 0 0 0 0 0 0 **5** 0 0 0 **1** 0 0 0 0 0 0 0	0 **2** 0 0 0 0 **1 2** 0 0 0 0 0 **1** 0 **7** 0 0 0 0 **2** 0 0 0 0 0 0
$3A_1$ $2A_0$ $A+2B+C_2$		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **8 9 4** 0 0 **2** 0 0 0 0 0 0	0 0 0 0 0 0 0 **2** 0 0 0 0 0 0 0 **5** 0 0 0 **1** 0 0 0 0 0 0 0	0 **2** 0 0 0 0 **1 2** 0 0 0 0 0 **1** 0 **7** 0 0 0 0 **2** 0 0 0 0 0 0
$3A_1$ $A+BB_0$ $A+2B+C_2$		
0 0 0 0 0 0 0 0 0 0 0 0 **2 1** 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 **2** 0 0 0 0 **2 1** 0 **5** 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 **1 2** 0 0 0 0 **2 1** 0 **7** 0 0 0 0 0 0 0 0 0 0 0
$3A_1$ $2A_0$ $2C_2$ (1)		
0 0 0 0 0 0 0 0 0 0 0 0 **2 1** 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 **2** 0 0 0 0 **2 1** 0 **5** 0 0 0 0 0 0 0 0 0 **1** 0	0 0 0 0 0 0 **1 2** 0 0 0 0 **2 1** 0 **7** 0 0 0 0 0 0 0 0 0 0 0
$3A_1$ $A+BB_0$ $2C_2$ (1)		
0 0 0 0 0 0 0 0 0 0 0 0 **2 1** 0 **8 9 4** 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 **2** 0 0 0 0 **2 1** 0 **5** 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 **1 2** 0 0 0 0 **2 1** 0 **7** 0 0 0 0 0 0 0 0 0 0 0
$3A_1$ $2BB_0$ $2C_2$ (1)		



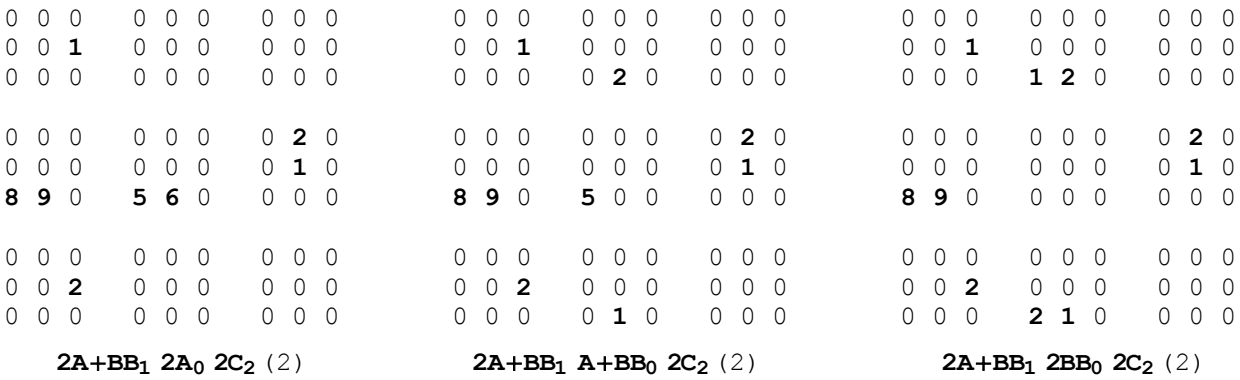
$3A_1\ 2A_0\ 2C_2$ (2) **$3A_1\ A{+}BB_0\ 2C_2$** (2) **$3A_1\ 2BB_0\ 2C_2$** (2)

$2A{+}BB_1\ 2A_0\ 2A{+}BB_2$ **$2A{+}BB_1\ A{+}BB_0\ 2A{+}BB_2$** **$2A{+}BB_1\ 2BB_0\ 2A{+}BB_2$**

$2A{+}BB_1\ 2A_0\ A{+}2BB_2$ **$2A{+}BB_1\ A{+}BB_0\ A{+}2BB_2$** **$2A{+}BB_1\ 2BB_0\ A{+}2BB_2$**

$2A{+}BB_1\ 2A_0\ A{+}2B{+}C_2$ **$2A{+}BB_1\ A{+}BB_0\ A{+}2B{+}C_2$** **$2A{+}BB_1\ 2BB_0\ A{+}2B{+}C_2$**

$2A{+}BB_1\ 2A_0\ 2C_2$ (1) **$2A{+}BB_1\ A{+}BB_0\ 2C_2$** (1) **$2A{+}BB_1\ 2BB_0\ 2C_2$** (1)



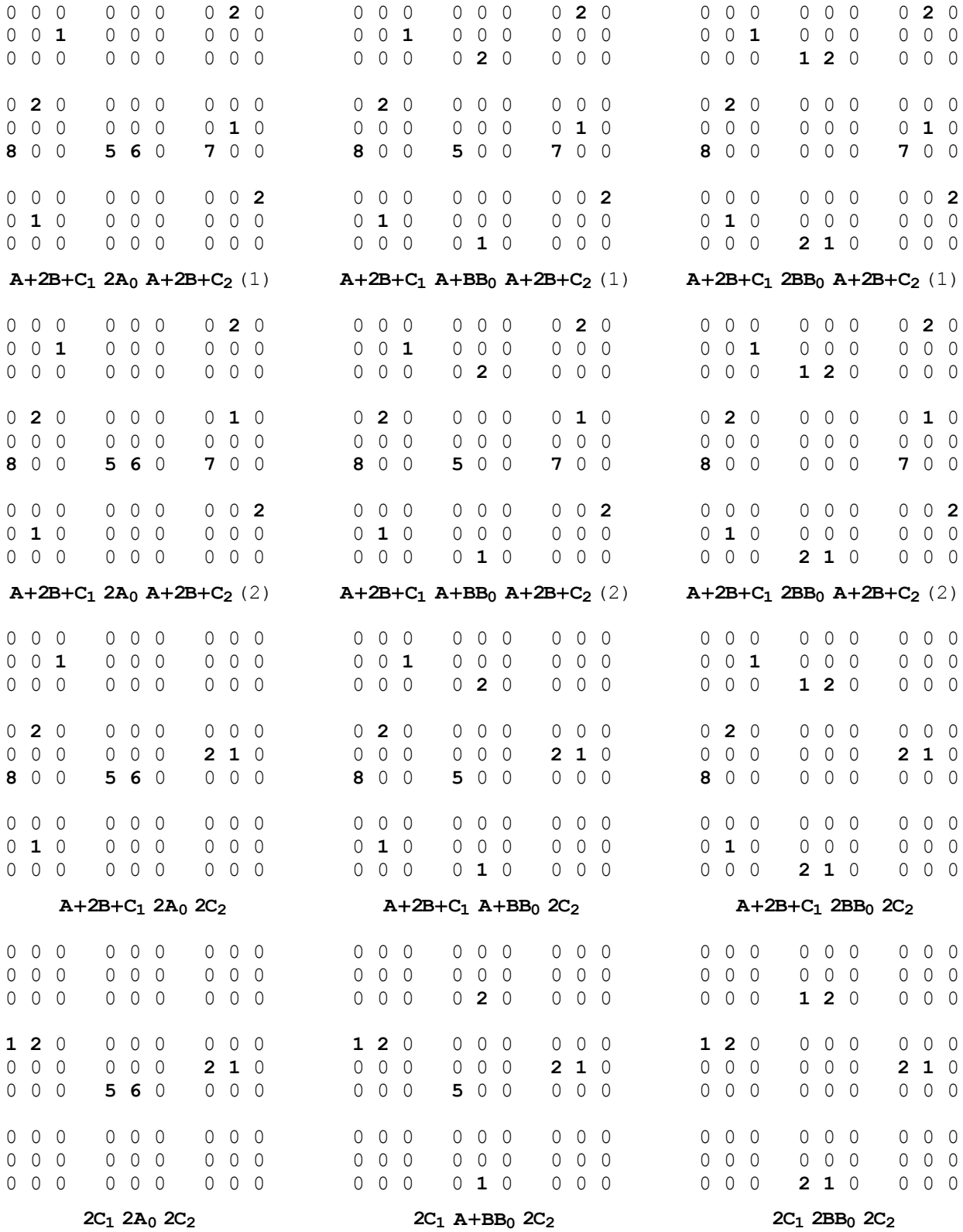
2A+BB$_1$ 2A$_0$ 2C$_2$ (2) **2A+BB$_1$ A+BB$_0$ 2C$_2$** (2) **2A+BB$_1$ 2BB$_0$ 2C$_2$** (2)

A+2BB$_1$ 2A$_0$ A+2BB$_2$ **A+2BB$_1$ A+BB$_0$ A+2BB$_2$** **A+2BB$_1$ 2BB$_0$ A+2BB$_2$**

A+2BB$_1$ 2A$_0$ A+2B+C$_2$ **A+2BB$_1$ A+BB$_0$ A+2B+C$_2$** **A+2BB$_1$ 2BB$_0$ A+2B+C$_2$**

A+2BB$_1$ 2A$_0$ 2C$_2$ (1) **A+2BB$_1$ A+BB$_0$ 2C$_2$** (1) **A+2BB$_1$ 2BB$_0$ 2C$_2$** (1)

A+2BB$_1$ 2A$_0$ 2C$_2$ (2) **A+2BB$_1$ A+BB$_0$ 2C$_2$** (2) **A+2BB$_1$ 2BB$_0$ 2C$_2$** (2)



Descartades la primera, en què l'interrogant pretenia significar la impossibilitat d'emplenar la casella 5,4 amb un valor que no fos **1** o **2** (que, una vegada ocupada, deixaria via lliure a **ACTUALITZA- PLUS** en l'emplenament automàtic de la següent, 6,4, amb el valor **2** o **1**) i les tres últimes, en què la presència dels valors **1** i **2** únicament en l'element de **R-L** corresponent a la casella 6,4 infringeix la **norma 3** a nivell de requadre 3×3 (l'existència dels 6, 7 o 8 elements no nuls restants, amb només 5, 6 o 7 valors, respectivament, seria la inevitable i complementària infracció a la **norma 2**), també podem excloure de l'estudi les tres combinacions que precedeixen aquestes, **A+2B+C1 2A0 2C2**, **A+2B+C1 A+BB0 2C2** i **A+2B+C1 2BB0 2C2**, on **ACTUALITZA- PLUS** entraria en acció directament per omplir amb un **2** el forat 6,4, en ser l'única casella del requadre central que admet aquest valor. De les altres, allò que més ens interessa ara dels elements de **R-L** corresponents al requadre 3×3 central és la presència de **1** i **2** en les llistes, raó per la qual seguirem amb la

notació usada fins ara, quant a representació de **nil** mitjançant **0**, però reservarem els caràcters en negreta per destacar-hi aquests valors. Surten 12 configuracions diferents, que detallem indicant quines són les combinacions que les comparteixen.

Compartides per $\mathbf{3A_1\ 2A_0\ 2A{+}BB_2}$, $\mathbf{3A_1\ 2A_0\ A{+}2BB_2}$, $\mathbf{2A{+}BB_1\ 2A_0\ 2A{+}BB_2}$, $\mathbf{2A{+}BB_1\ 2A_0\ A{+}2BB_2}$ i $\mathbf{A{+}2BB_1\ 2A_0\ A{+}2BB_2}$:

(**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9)

(**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9)

0 0 (**1 2** 3 4 0 0 7 8 9)

Compartides per $\mathbf{3A_1\ A{+}BB_0\ 3A_2}$, $\mathbf{3A_1\ A{+}BB_0\ 2A{+}BB_2}$, $\mathbf{3A_1\ A{+}BB_0\ A{+}2BB_2}$, $\mathbf{2A{+}BB_1\ A{+}BB_0\ 2A{+}BB_2}$, $\mathbf{2A{+}BB_1\ A{+}BB_0\ A{+}2BB_2}$ i $\mathbf{A{+}2BB_1\ A{+}BB_0\ A{+}2BB_2}$:

(**1 2** 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

(**1 2** 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

0 (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

Compartides per $\mathbf{3A_1\ 2BB_0\ 3A_2}$, $\mathbf{3A_1\ 2BB_0\ 2A{+}BB_2}$, $\mathbf{3A_1\ 2BB_0\ A{+}2BB_2}$, $\mathbf{2A{+}BB_1\ 2BB_0\ 2A{+}BB_2}$, $\mathbf{2A{+}BB_1\ 2BB_0\ A{+}2BB_2}$ i $\mathbf{A{+}2BB_1\ 2BB_0\ A{+}2BB_2}$:

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

(0 0 3 0 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

Compartides per $\mathbf{3A_1\ 2A_0\ A{+}2B{+}C_2}$, $\mathbf{2A{+}BB_1\ 2A_0\ A{+}2B{+}C_2}$ i $\mathbf{A{+}2BB_1\ 2A_0\ A{+}2B{+}C_2}$:

(**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9)

(0 **2** 3 4 0 0 7 8 9) (0 **2** 3 4 0 0 7 8 9) (0 **2** 3 4 0 0 7 8 9)

0 0 (**1 2** 3 4 0 0 7 8 9)

Compartides per $\mathbf{3A_1\ A{+}BB_0\ A{+}2B{+}C_2}$, $\mathbf{2A{+}BB_1\ A{+}BB_0\ A{+}2B{+}C_2}$ i $\mathbf{A{+}2BB_1\ A{+}BB_0\ A{+}2B{+}C_2}$:

(**1 2** 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

(0 **2** 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (0 **2** 3 4 0 6 7 8 9)

0 (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

Compartides per $\mathbf{3A_1\ 2BB_0\ A{+}2B{+}C_2}$, $\mathbf{2A{+}BB_1\ 2BB_0\ A{+}2B{+}C_2}$ i $\mathbf{A{+}2BB_1\ 2BB_0\ A{+}2B{+}C_2}$:

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (0 **2** 3 4 5 6 7 8 9)

(0 0 3 0 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

Compartides per $\mathbf{3A_1\ 2A_0\ 2C_2}$(1), $\mathbf{2A{+}BB_1\ 2A_0\ 2C_2}$(1), $\mathbf{A{+}2BB_1\ 2A_0\ 2C_2}$(1) i $\mathbf{A{+}2B{+}C_1\ 2A_0\ A{+}2B{+}C_2}$(2):

(**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9) (**1 2** 3 4 0 0 7 8 9)

(0 0 3 4 0 0 7 8 9) (0 0 3 4 0 0 7 8 9) (0 0 3 4 0 0 7 8 9)

0 0 (**1 2** 3 4 0 0 7 8 9)

Compartides per **$3A_1$ A+BB_0 $2C_2$** (1), **2A+BB_1 A+BB_0 $2C_2$** (1), **A+$2BB_1$ A+BB_0 $2C_2$** (1) i **A+2B+C_1 A+BB_0 A+2B+C_2** (2):

(**1 2** 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

(0 0 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9)

0 (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

Compartides per **$3A_1$ $2BB_0$ $2C_2$** (1), **2A+BB_1 $2BB_0$ $2C_2$** (1), **A+$2BB_1$ $2BB_0$ $2C_2$** (1) i **A+2B+C_1 $2BB_0$ A+2B+C_2** (2):

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9)

(0 0 3 0 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

Compartides per **$3A_1$ $2A_0$ $2C_2$** (2), **2A+BB_1 $2A_0$ $2C_2$** (2), **A+$2BB_1$ $2A_0$ $2C_2$** (2) i **A+2B+C_1 $2A_0$ A+2B+C_2** (1):

(**1** 0 3 4 0 0 7 8 9) (**1** 0 3 4 0 0 7 8 9) (**1** 0 3 4 0 0 7 8 9)

(0 **2** 3 4 0 0 7 8 9) (0 **2** 3 4 0 0 7 8 9) (0 **2** 3 4 0 0 7 8 9)

0 0 (**1 2** 3 4 0 0 7 8 9)

Compartides per **$3A_1$ A+BB_0 $2C_2$** (2), **2A+BB_1 A+BB_0 $2C_2$** (2), **A+$2BB_1$ A+BB_0 $2C_2$** (2) i **A+2B+C_1 A+BB_0 A+2B+C_2** (1):

(**1** 0 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (**1** 0 3 4 0 6 7 8 9)

(0 **2** 3 4 0 6 7 8 9) (0 0 3 4 0 6 7 8 9) (0 **2** 3 4 0 6 7 8 9)

0 (0 0 3 4 0 6 7 8 9) (**1 2** 3 4 0 6 7 8 9)

Compartides per **$3A_1$ $2BB_0$ $2C_2$** (2), **2A+BB_1 $2BB_0$ $2C_2$** (2), **A+$2BB_1$ $2BB_0$ $2C_2$** (2) i **A+2B+C_1 $2BB_0$ A+2B+C_2** (1):

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1** 0 3 4 5 6 7 8 9)

(0 0 3 4 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (0 **2** 3 4 5 6 7 8 9)

(0 0 3 0 5 6 7 8 9) (0 0 3 4 5 6 7 8 9) (**1 2** 3 4 5 6 7 8 9)

El pas següent consistia a descartar les combinacions no conflictives en cap de les interpretacions possibles relatives a l'ordre d'emplenament de les caselles (no conflictives perquè la infracció de les **normes 2/3** en explorar la 4ª fila ja es detectés explorant el requadre 3×3 central, o perquè es produïssin emplenaments automàtics a càrrec d'**ACTUALITZA-PLUS** que evitessin la infracció), eliminant també les configuracions **R-L** corresponents al requadre 3×3 central de combinacions totes les quals haguessin estat descartades. Llavors venia la decissió més compromesa, perquè havíem de definir un perfil que fos comú a totes les configuracions que haguessin sobreviscut però que no fos tan folgat que en deixés entrar moltes més: no oblideu que tota aquesta història (diagnosticar indirectament infraccions a nivell de fila o columna, detectant determinades característiques dels requadres 3×3) la muntàvem per evitar-nos ampliar l'acció d'**EXPLORA-3*9** a totes les files i totes les columnes. Doncs bé: no ha calgut ni coronar la primera d'aquestes etapes perquè exactament en la combinació **2A+BB_1 $2A_0$ A+$2BB_2$** han començat a passar coses estranyes, seguint la pista de les quals hem vist que la clau de volta d'aquests conflictes no residia a **EXPLORA-3*9** sinó a **ACTUALITZA-PLUS**. La descoberta no ha estat ni de bon tros una il·luminació instantània, raó per la qual val la pena que descrivim amb cert detall les experiències que s'anaven succeint i les conclusions provisionals que en trèiem, fins que gairebé per reducció a l'absurd la solució més raonable ha brillat obrint-se pas per damunt de les demés.

```
0 0 0   0 0 0   0 2 1     0 0 0   0 0 0   0 2 0     6 5 9   8 7 3   1 2 4
0 0 1   0 0 0   0 0 0     0 0 1   0 0 0   0 0 0     7 4 1   9 2 5   6 3 8
0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     3 2 8   1 6 4   9 7 5

0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     4 7 6   3 9 8   2 5 1
0 0 0   0 0 0   0 0 0     0 0 0   0 0 0   0 0 0     2 1 5   7 4 6   8 9 3
8 9 0   5 6 0   7 0 0     8 9 3   5 1 6   7 4 X     8 9 3   5 1 2   7 4 6

0 0 0   0 0 0   0 1 2     5 6 7   4 8 9   3 1 2     5 6 7   4 8 9   3 1 2
0 0 2   0 0 0   0 0 0     1 3 2   6 5 7   4 8 9     1 3 2   6 5 7   4 8 9
0 0 0   0 0 0   0 0 0     9 8 4   1 3 2   5 6 7     9 8 4   2 3 1   5 6 7
```

A l'esquerra teniu la combinació **2A+BB$_1$ 2A$_0$ A+2BB$_2$**, on cal farcir el requadre 9×3 inferior per evitar que permuti la seva posició amb el del mig i, en quedar la 4ª línia (que és la sospitosa) en posició inferior, no arribem a percebre que alguns candidats ens fan esperar massa. En el centre ja ho hem arreglat emplenant amb els valors **8**, **9**, **4** i **3** algunes caselles de la 1ª línia, però el conflicte no sorgeix amb el candidat **1** a 9,9 (quan la casella 5,4 ja està ocupada pel **6**) o amb el **6** a 5,4 (quan la casella 9,9 ja està ocupada per l'**1**), sinó quan cap de les caselles esmentadas no està ocupada encara i justament volem emplenar la 5,4, activant-la: si fem això, el caseller central 3×3 quedarà congelat amb el text habitual *Espereu el missatge "CLIC sobre el nombre..."*, perquè el veredicte sobre el candidat **1** mai no acaba d'arribar. En aquesta mateixa representació hem anotat els valors que són assignats en la primera embranzida de **RE-MEMBER**, que queda deturada a la casella 9,4 (X). A la dreta hem representat l'emplenament total de l'escaquer 9×9 amb el mateix candidat a 5,4: observeu que la clau de l'exit és a les caselles 4,1 i 6,1, que en comptes d'adoptar virtualment els valors 1 i 2, respectivament, han adoptat els valors 2 i 1; com ja estem acostumats a veure, la regressió des de 9,4 a 4,1 és la responsable que l'espera sigui tan llarga. Però no us precipiteu a dir "és clar que sí, perquè si ens estem trencant les banyes, tractant d'identificar les configuracions de les línies (o columnes) i requadres 3×3 que acompanyen aquesta problemàtica, és precisament per evitar aquestes llargues esperes", perquè aquí el cas és una mica diferent. Només cal veure quin és el fragment de **(cadr 2R)** (**R-LLL** actualitzat) corresponent a la 4ª línia, quan emplenem virtualment 5,4 amb l'**1**:

```
0 0 (0 0 3 0 0 6 0 0 0) 0 0 (0 2 3 4 0 6 0 0 0) 0 (0 0 3 4 0 6 0 0 0) (0 0 3 4 0 6 0 0 0)
```

Hem escrit el **2** en negreta perquè sigui més fàcil adonar-se que aquesta aparició és única en la fila (no pas en el requadre 3×3 central). El **2** i únicament el **2**: ben diferent del que cercàvem fins ara, que era la aparició única de dos valors en una d'aquestes llistes. No us estranyi si us quedeu astorats de moment, com li va passar a l'autor, i tot seguit dieu "però si això no tenia que haver passat perquè **ACTUALITZA-PLUS** era l'encarregat de tapar aquests forats", perquè és la reacció lògica de qui recorda què hem anat fent fins ara però ja no recorda tan bé com ho hem fet. Per sortir de dubtes, aneu a l'última versió d'**ACTUALITZA-PLUS** i quedareu amb un pam de nas: el rastreig de llistes en què sortís un únic valor l'havíem fet casella a casella, però el de conjunts de caselles per veure si algun valor només sortia en una llista del conjunt únicament s'havia fet per a un tipus d'unitat, el requadre 3×3; igual que a **EXPLORA-3*9**, no havíem cercat per files ni per columnes. Un altra cop caldrà arranjar una badada però, com que carregar amb més feina una funció omnipresent com **ACTUALITZA-PLUS** podria desembocar fàcilment en un augment notable dels temps d'espera, convindrà filar prim. La primera cosa que descobrim quan repassem l'àmbit d'actuació d'aquesta funció és que havíem omès l'exploració per files i columnes (a hores d'ara l'autor ja no recorda si l'omissió havia estat deliberada, per creure que ja n'hi havia prou a explorar els requadres 3×3, i ara ens hem trobat que aquesta pressumpció era falsa) i tanmateix ens havíem excedit a l'estendre l'exploració als 9 requadres que componen l'escaquer: només amb els 5 que comparteixen fila i columna amb la casella actual n'hi hauria prou; pel que fa a files i columnes, d'entrada sembla que no caldria abastar-les totes (9+9) i que n'hi hauria prou amb les 3+3 en què podem descompondre els 5 requadres esmentats:

```
(defun PLUS-N->P (M / P1 P2)
  (if (= K 1)
      (progn
        (setq A-N N A-P M PLUS T)
        (if REAL
            (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))
            (if INI (setq REPLUS (ACT-9*9 N M REPLUS)))))))
```

```
; Exploració per 5 requadres 3*3, 3 files i 3 columnes.
(defun ACTUALITZA-PLUS (/ 3*3 LL3 M N X1 X2 Y1 Y2 5P1 TRANSP)
  (mapcar '(lambda (F-9*9 F-LLL)
             (if (not PLUS)
                 (mapcar '(lambda (E-9*9 E-LLL)
                            (if (and (not PLUS) E-LLL)
                                (progn
                                  (setq K 0)
                                  (foreach E L1A9
                                    (if (and (< K 2) (member E E-LLL))
                                        (setq N E K (1+ K))))
                                  (if (= K 1) (PLUS-N->P E-9*9)))))
                          F-9*9 F-LLL)))
          A-9*9 A-LLL)
  (if (not PLUS)
      (progn
        (setq X2 (* (/ X 3) 3) Y2 (* (/ Y 3) 3))
        (foreach K '(6 3 0) (if (/= K Y2) (setq 5P1 (cons (list X2 K) 5P1))))
        (foreach K '(6 3 0) (setq 5P1 (cons (list K Y2) 5P1)))
        (foreach P1 5P1
          (if (not PLUS)
              (progn
                (setq X1 (car P1) Y1 (cadr P1)
                      3*3 (SUB-X*Y A-9*9 P1 (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL P1 (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0)
                        (mapcar '(lambda (F-3*3 F-LL3)
                                   (mapcar '(lambda (E-3*3 E-LL3)
                                              (if (member N E-LL3)
                                                  (setq K (1+ K) M E-3*3)))
                                           F-3*3 F-LL3))
                                3*3 LL3)
                        (PLUS-N->P M)))))))))
  (repeat 2
    (if (not PLUS)
        (progn
          (if TRANSP
              (setq 3*3 (TRANSPOSAR (SUB-X*Y A-9*9 (list X2 0) (list (+ X2 2) 8)))
                    LL3 (TRANSPOSAR (SUB-X*Y A-LLL (list X2 0) (list (+ X2 2) 8))))
              (setq 3*3 (SUB-X*Y A-9*9 (list 0 Y2) (list 8 (+ Y2 2)))
                    LL3 (SUB-X*Y A-LLL (list 0 Y2) (list 8 (+ Y2 2)))))
          (foreach N L1A9
            (if (not PLUS)
                (mapcar '(lambda (F-9*3 F-LL3)
                           (setq K 0)
                           (mapcar '(lambda (E-9*3 E-LL3)
                                      (if (member N E-LL3)
                                          (setq K (1+ K) M E-9*3)))
                                   F-9*3 F-LL3)
                           (PLUS-N->P M))
                        3*3 LL3)))
          (setq TRANSP T)))))
```

Tanmateix no hem trigat a descobrir que havíem dut massa lluny el zel estalviador: estava bé limitar l'exploració als 5 requadres esmentats, però calia explorar les 9 files i les 9 columnes, amb independència d'on se situés la casella a emplenar. Per no allargar-ho encara més, n'hi a prou a considerar una petita variació sobre el cas que ha aixecat la llebre: si en les dues representacions precedents (centre o dreta) haguéssim considerat que la casella 5,4 (**1**) ja estava plena i era 8,9 la que anàvem a emplenar, tot hauria anat igual amb el primer candidat (**2**), tot i que la conflictiva 4ª línia ja quedava fora dels criteris d'exploració incorporats a l'última versió d'**ACTUALITZA-PLUS**. Per aquest motiu en presentem una de nova (sols la tercera part, i com que **PLUS-N->P** no varia respecte l'última, no la reproduïm):

```
; Exploració per 5 requadres 3*3, 9 files i 9 columnes.
(defun ACTUALITZA-PLUS (/ 3*3 LL3 M N X1 X2 Y1 Y2 5P1)
  (mapcar '(lambda (F-9*9 F-LLL)
             ...................
          A-9*9 A-LLL)
  (if (not PLUS)
      (progn ...)
  (if (not PLUS)
      (mapcar '(lambda (9*9 LL3)
                 (foreach N L1A9
                   (if (not PLUS)
                       (mapcar '(lambda (F-9*9 F-LL3)
                                  (setq K 0)
                                  (mapcar '(lambda (E-9*9 E-LL3)
                                             (if (member N E-LL3)
                                                 (setq K (1+ K) M E-9*9)))
                                          F-9*9 F-LL3)
                                  (PLUS-N->P M))
                               9*9 LL3))))
              (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL)))))
```

Sortosament, comprovem que la retallada de 4 requadres 3×3 haurà compensat, si fa no fa, la inclusió de 9 files i 9 columnes perquè els temps gairebé no han variat. A partir d'aquí, sorgeixen aquests dubtes:

- Amb la correcció d'**ACTUALITZA-PLUS**, ¿haurà desaparegut la problemàtica de les infraccions a les **normes 2** i **3** a nivell de files o columnes que romanen ocultes quan limitem la inspecció als requadres 3×3?
- Si no és així i encara perviu la possibilitat que en el fragment de **(cadr 2R)** corresponent a una fila o columna hi hagi aparicions úniques de diversos valors 1... 9 concentrats en una mateixa llista, ¿podríem integrar el dispositiu *ad hoc* en aquesta funció, en lloc de fer-ho a **EXPLORA-3*9**, ara que s'hi exploren totes les files i columnes?

Pel que fa a la primera qüestió, podem provar algunes de les combinacions que sols reixien amb la versió **EXPLORA-3*9** + **2*FIL/COL** (**3A$_1$ 2BB$_0$ 2C$_2$**(1) i **3A$_1$ 2BB$_0$ 2C$_2$**(2), que corresponen als casos que havien fet saltar l'alarma) o que encara s'hi resistien (l'últimament estudiada **2A+BB$_1$ 2A$_0$ A+2BB$_2$**) per veure si responen al nou tractament. Òbviament, per avaluar l'acció de la nova **ACTUALITZA-PLUS** hem abandonat la versió esmentada d'**EXPLORA-3*9**, tornant a la precedent.

0 0 0 0 0 **1** 0 0 0	0 0 0 0 0 0 0 0 0	0 **2** 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 **1 2 3**	0 0 0 0 0 0 0 0 0	0 0 0 0 0 **1** 0 0 0	0 0 0 0 0 0 0 **2** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **8 9** 0	0 0 0 0 0 0 **5 6** 2	0 0 0 0 0 0 **7** 4 1	0 0 0 0 0 0 **8 9 4**	0 0(1) 0 0 0 0 0 0 2	0(**2**)0 (**2**)**1** 0 0 0 0	0 0 0 0 0 0 **8 9** 0	0 0 0 0 0 0 **5** 0 0	0 0 0 **2 1** 0 0 0 0
0 0 0 0 0 **2** **9 8 4**	0 0 0 0 0 0 2 **3** 1	0 **1 2** 0 0 0 0 0 0	0 0 0 0 0 0 **3 6 7**	0 0 0 **9 4 8** **2 1** 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 **2** **9 8 4**	0 0 0 0 0 0 0 **1** 0	0 0 0 0 **8 9** 0 0 0

A l'esquerra, teniu de nou la combinació **2A+BB$_1$ 2A$_0$ A+2BB$_2$** amb el farciment precís perquè no es permutin els requadres 9×3, sols que ara l'**ACTUALITZA-PLUS** que també escruta línies i columnes ja no ens farà esperar tant de temps els veredictes de compatibilitat dels candidats **1**, **2**, **4** i **6** a la casella 5,4 sinó que els presentarà en un tres i no res. Si li assignem l'últim valor, com es veu a la representació (fins aquí, tots els valors hi figuren en negreta), **ACTUALITZA** (**REAL** = **T**) ens hi abocarà tots els valors amb què poc abans **ACTUALITZA-PLUS** havia anat tapant forats virtualment (**REAL** = **nil**): l'ordre d'aquesta reacció en cadena (valors en caràcters normals) és 4 (8,4), 1 (9,4), 2 (6,4), 2 (4,1) i 1 (6,1); queda clar que l'1 a 9,4 impedirà que aquest mateix valor pugui ser candidat a la casella 9,9. Pel que fa a **3A$_1$ 2BB$_0$ 2C$_2$** (1 i 2), que hem representat conjuntament en el centre per no ocupar més espai (en el requadre central-dret, la casella ocupada amb un **2** és 7,5 en un cas i 8,6 en l'altre, i aquests valors s'han posat entre parèntesis), quan a 8,5 provem el candidat **1** tampoc es farà esperar gens el veredicte d'incompatibilitat, perquè en ambdós casos la nova **ACTUALITZA-PLUS** haurà col·locat efectivament un 2 a 6,4 i, després d'això, en emplenar virtualment 8,5 amb el candidat **1**, col·locarà

virtualment el valor (1) a 6,6, i això comportarà una infracció a la **norma 1** en el requadre 3×3 central-esquerre (on el valor 1 no el tenim a **R-9** ni a **R-L**, per bé que en el primer cas el valor 2 aparegui en el 3r tercet i en el segon en el 2n). En la primera combinació **ACTUALITZA-PLUS** salva la situació ella soleta, mentre que a la segona i tercera es limita a preparar el terreny i és **EXPLORA-3*9** qui dóna el cop de gràcia, però en tots tres casos ens quedem sense cap resposta concloent a la pregunta que ens fèiem. Haurem d'arribar a la combinació **2A+BB$_1$ A+BB$_0$ 2C$_2$** (1) per veure com l'acció conjunta d'ambdues funcions (tal i com són ara) és insuficient, perquè encara es poden produir casos en què la infracció de les **normes 2**/**3** en una línia o columna no quedi registrada explorant per requadres 3×3 i ocasioni esperes inadmissibles en el veredicte de compatibilitat d'un valor candidat. La teniu a la dreta de les combinacions comentades, a la pàgina precedent, amb farciment perquè no es permutin els requadres 9×3, i per provocar la situació conflictiva que volem caldrà seguir un ordre determinat d'emplenaments, abans que li arribi el torn a la casella 4,4 (**5**): no heu d'emplenar 5,1 (**1**) i 5,7 (**2**) (ni 4,4 (**5**), que cal deixar per al final, ho repetim) sense tenir prèviament ocupades (valors en negreta) la resta de caselles; altrament, emplenant per exemple 5,1 (**1**) i 4,4 (**5**) abans que 5,7, ja no hi haurà ocasió d'assignar-hi un **2**, perquè **ACTUALITZA-PLUS** haurà actuat emplenant 6,4 (1) i 5,4 (2). Fent-ho com hem dit, el resultat d'activar la casella 4,4 serà la ràpida aparició dels candidats 1 i 2 i un prolongat silenci de quatre minuts que no en valida cap més: el veredicte d'incompatibilitat dels 5 candidats següents (3, 4, el nostre **5**, 6 i 7) triga gairebé 50 segons cada un. La raó és que l'exploració per requadres 3×3 d'**EXPLORA-3*9** no ha detectat cap infracció de les **normes 2**/**3**, però sí que n'hi ha en la 4ª línia, on en el fragment de **(cadr 2R)** corresponent **1** i **2** només surten a la llista (**1 2** 3 4 0 6 7 0 0) de la casella 6,4.

Això ens permet passar a la segona de les preguntes que ens formulàvem a la pàgina precedent. Tot el desplegament exhaustiu de combinacions que fèiem tenia un únic propòsit: perfilar el repertori de configuracions **R-L** de requadres 3×3 (repertori que, després de millorar **ACTUALITZA-PLUS**, s'haurà reduït considerablement) en què poden aparèixer files o columnes que incompleixin les **normes 2**/**3**, provisionalment limitat al cas particular de dos valors que només apareguin en la mateixa llista. Recordarem que aquest circumloqui només estava justificat per l'obsessió d'evitar estendre l'exploració a totes les files i columnes, i en totes les intervencions d'**EXPLORA-3*9**, i encara quedava per veure si aquesta tàctica era viable: bé podia passar que no hi hagués cap correspondència codificable en termes senzills (és a dir, que el perfil es reduís a una enumeració exhaustiva de configuracions) i que això encara portés més feina que abordar sempre una exploració desacomplexada per files i columnes. Però ara que hem descobert la necessitat d'estendre l'exploració d'**EXPLORA-3*9** a totes les files i columnes, el dilema desapareix, perquè costarà ben poc aprofitar aquesta dinàmica per introduir-hi una verificació directa. Només caldrà sumar a la variable **PLUS** (local a **ACTUALITZA**, on s'inicialitza a **nil**) una altra variable **FORA** (homónima de la d'**EXPLORA-3*9** i semànticament similar, en la mesura que avorta una línia d'emplenaments virtuals, però no definida en aquesta funció sinó en l'àmbit més extens de **RE-MEMBER**), processant les files i columnes mentre sigui **(not PLUS)** i **(not FORA)**, i tanmateix esgotant els valors **N** (1... 9) mentre **(not FORA)** (encara que **PLUS** hagi estat activat), de la manera següent:

- Si en una línia (fila o columna) de **(cadr 2R)** detecta que només una llista té **N**, guarda les coordenades de la casella corresponent (emmagatzemades a **(car 2R)**) a la llista **5P1** (obsoleta després d'haver representat les coordenades definidores dels 5 requadres 3×3 que comparteixen fila i columna amb la casella actual) i, si encara **PLUS** = **nil**, assigna **N** a la casella i activa **PLUS** mitjançant **PLUS-N->P**.

Fins aquí tot seria gairebé igual al que fèiem fins ara, però hi ha dues novetats:

- Si en una línia hi ha algun valor **N** que no aparegui a **(car 2R)** ni a **(cadr 2R)**, activarem **FORA**: ja que no donem per suficient l'exploració per requadres 3×3 pel que fa a les **normes 2**/**3**, tampoc no la donarem pel que fa a la **norma 1**.
- Si, a més del valor **N$_1$**, en la mateixa línia n'ha sortit algun altre **N$_2$** que sols figura en una llista:
  - Si **N$_1$** i **N$_2$** comparteixen la mateixa llista, som en el cas d'infracció de les **normes 2**/**3** que fa estona que ens fa anar de corcoll, i també activarem **FORA**.
  - Si **N$_1$** i **N$_2$** se situen en llistes diferents, no farem res: com que **PLUS** ja s'ha activat, en una pròxima passada per **ACTUALITZA-PLUS** s'assignarà aquest valor a la casella que correspongui (en explorar per requadres 3×3, files o columnes).

Veiem com hauran quedat **PLUS-N->P**, **ACTUALITZA-PLUS** i **ACTUALITZA** (en què només hem ampliat la condició **(not PLUS)** a **(not (or PLUS FORA))**) després dels últims canvis. Tornem a incloure la funció **TERCETS** perquè hem aconseguit compactar-la una mica en relació a l'última versió. Pel que fa a **EXPLORA-3*9** retornem a la penúltima, si bé

mantenim l'exclusió dels requadres 3×3 amb les 9 caselles ocupades, estalviant-los passar per **EXPL-SUBCONJ** i **TERCETS**, raó per la qual també la incloem. Com és lògic, si la variable **FORA** que s'activa a **ACTUALITZA-PLUS**, homònima de la que és local en **EXPLORA-3*9**, ho ha de ser a **RE-MEMBER**, caldrà modificar mínimament aquesta funció: per destacar-ne les modificacions, les hem subratllat.

```
(defun PLUS-N->P (M / P1 P2)
  (setq A-N N A-P M PLUS T)
  (if REAL
      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N))
      (if INI (setq REPLUS (ACT-9*9 N M REPLUS)))))

; Exploració per 5 requadres 3*3, 9 files i 9 columnes (en què també es mira si hi
; ha 2 o més valors admissibles que només apareixen 1 cop i a la mateixa casella).
(defun ACTUALITZA-PLUS (/ 3*3 LL3 M N X1 X2 Y1 Y2 P1 5P1)
  (mapcar '(lambda (F-9*9 F-LLL)
             (if (not PLUS)
                 (mapcar '(lambda (E-9*9 E-LLL)
                            (if (and (not PLUS) E-LLL)
                                (progn
                                  (setq K 0)
                                  (foreach E L1A9
                                    (if (and (< K 2) (member E E-LLL))
                                        (setq N E K (1+ K))))
                                  (if (= K 1) (PLUS-N->P E-9*9)))))
                         F-9*9 F-LLL)))
           A-9*9 A-LLL)
  (if (not PLUS)
      (progn
        (setq X2 (* (/ X 3) 3) Y2 (* (/ Y 3) 3))
        (foreach K '(6 3 0) (if (/= K Y2) (setq 5P1 (cons (list X2 K) 5P1))))
        (foreach K '(6 3 0) (setq 5P1 (cons (list K Y2) 5P1)))
        (foreach P1 5P1
          (if (not PLUS)
              (progn
                (setq X1 (car P1) Y1 (cadr P1)
                      3*3 (SUB-X*Y A-9*9 P1 (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL P1 (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0)
                        (mapcar '(lambda (F-3*3 F-LL3)
                                   (mapcar '(lambda (E-3*3 E-LL3)
                                              (if (member N E-LL3)
                                                  (setq K (1+ K) M E-3*3)))
                                           F-3*3 F-LL3))
                                3*3 LL3)
                        (if (= K 1) (PLUS-N->P M)))))))))
  (setq FORA ())
  (mapcar '(lambda (9*9 LL3)
             (if (not (or PLUS FORA)) ; Equival a (and (not PLUS) (not FORA))
                 (mapcar '(lambda (F-9*9 F-LL3)
                            (if (not (or PLUS FORA))
                                (progn
                                  (setq 5P1 ())
                                  (foreach N L1A9
                                    (if (not FORA)
                                        (progn
                                          (setq K 0)
                                          (mapcar '(lambda (E-9*9 E-LL3)
                                                     (if (member N E-LL3)
                                                         (setq K (1+ K) M E-9*9)))
                                                  F-9*9 F-LL3)
                                          (if (< K 2)
                                              (if (= K 0)
                                                  (if (not (member N F-9*9))
                                                      (setq FORA T))
```

```
                                                    (progn
                                                      (if (not PLUS) (PLUS-N->P M))
                                                      (setq 5P1 (cons M 5P1))))))))
                                        (if (and (not FORA) (> (length 5P1) 1))
                                            (while (setq K (car 5P1) 5P1 (cdr 5P1))
                                              (if (member K 5P1) (setq FORA T)))))))
                                 9*9 LL3)))
          (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L PLUS)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
              A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM"
                           "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
      (while (not (or PLUS FORA)) ; Equival a (and (not PLUS) (not FORA))
             (setq X (car A-P) Y (cadr A-P)
                   A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (ACT-LLL X Y A-LLL))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

(defun TERCETS (<R-9 <R-L / R-9 R-L TRANSP TERC SEGUIR I J K K1 K2)
  (foreach 3N 3*N
    (if (and (not TERC) (member N 3N))
        (progn
          (setq R-9 <R-9 R-L <R-L
                N1 (car 3N) N2 (cadr 3N) N3 (last 3N)
                TRANSP ())
          (repeat 2
            (if (not TERC)
                (progn
                  (setq I () J -1 K 0)
                  (mapcar '(lambda (F/R-9 F/R-L)
                             (setq J (1+ J) K1 0 K2 0)
                             (mapcar '(lambda (E/R-9 E/R-L)
                                        (if (and (or E/R-L (= N1 E/R-9)
                                                     (= N2 E/R-9) (= N3 E/R-9))
                                                 (or (not E/R-L) (member N1 E/R-L)
                                                     (member N2 E/R-L)
                                                     (member N3 E/R-L)))
                                            (setq K1 (1+ K1))
                                            (setq K2 (1+ K2))))
                                      F/R-9 F/R-L)
                             (if (= K1 3) (setq I J))
                             (if (= K2 3) (setq K (1+ K))))
                           R-9 R-L)
                  (if (and I (= K 2))
                      (progn
                        (foreach L (setq J (nth I R-L))
                          (setq K1 ())
                          (foreach E (setq K L)
                            (if (and E (not (member E 3N)))
                                (setq TERC T K1 T K (subst () E K))))
                          (if K1 (setq J (subst K L J))))
                        (if TERC (setq R-L (subst J (nth I R-L) R-L)))
                        (setq J -1 K1 () SEGUIR T)
                        (foreach N (reverse L1A9)
                          (if (not (member N 3N)) (setq K1 (cons N K1))))
```

```
                        (while (and SEGUIR (< J 3))
                          (setq K2 K1 J (1+ J))
                          (if (/= J I)
                              (progn
                                (mapcar '(lambda (E/R-9 E/R-L)
                                           (if E/R-L
                                               (foreach N E/R-L
                                                 (if (member N K2)
                                                     (setq K2 (K2- N))))
                                               (setq K2 (K2- E/R-9))))
                                        (nth J R-9) (nth J R-L))
                                (if (= (length K2) 3)
                                    (progn
                                      (setq SEGUIR ())
                                      (foreach L (setq K (nth (- 3 (+ I J)) R-L)
                                                       K1 K)
                                        (if L
                                            (progn
                                              (setq I () J I)
                                              (foreach E L
                                                (setq J
                                                      (cons
                                                        (if E
                                                            (if (member E K2)
                                                                E
                                                                (not (setq I T))))
                                                        J)))
                                              (if I (setq K1 (subst (reverse J)
                                                                    L K1))))))
                                      (if (not (equal K1 K))
                                          (setq R-L (subst K1 K R-L) TERC T)))))))
                        (if TERC (setq R-L (if TRANSP (TRANSP-3*3 R-L) R-L)
                                       TN/R-L (3*3/R-L TN/R-L)))))
                  (if (not (or TRANSP TERC))
                      (setq R-9 (TRANSP-3*3 R-9)
                            R-L (TRANSP-3*3 R-L) TRANSP T)))))))))

(defun EXPLORA-3*9 (/ FORA R-9 R-L L LL I M) ; EXPLORACIÓ NOMÉS PER REQUADRES 3*3.
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y (car 2R) (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y (cadr 2R) (list X Y) (list (+ X 2) (+ Y 2))) L ())
            (foreach F R-9
              (foreach E F (if (atom E) (setq L (cons E L)))))
            (if (< (length L) 9)
                (progn
                  (foreach F R-L
                    (foreach E F
                      (if E (foreach M E
                              (if (and M (not (member M L)))
                                  (setq L (cons M L)))))))
                  (if (< (length L) 9) (setq FORA T))
                  (if (and INI (not FORA))
                      (progn
                        (setq L ())
                        (foreach F R-L
                          (foreach E F
                            (if E (progn
                                    (setq I ())
                                    (foreach M E (if M (setq I (cons M I))))
                                    (setq L (cons I L))))))
                        (EXPL-SUBCONJ ())
                        (if (not FORA) (TERCETS R-9 R-L))))))
            (if (and INI (not FORA)) (setq N/R-L (3*3/R-L N/R-L)))))))
  FORA)
```

```
(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ())
              INI (or FORA (EXPLORA-3*9)))
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
  (if (not (or INI OK))
      (if (not (setq OK (COMPLET-9*9 R-9*9)))
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (if FORA
                        (setq FORA ())
                        (if (equal (cadr 2R) NIL*NIL)
                            (setq OK T)
                            (progn
                              (foreach F (cadr 2R)
                                (if (and (not OK) (member (car NIL*NIL) F))
                                    (setq OK T)))
                              (if (not OK) (setq OK (EXPLORA-3*9)))
                              (if OK
                                  (setq OK ())
                                  (RE-MEMBER (car 2R) (cadr 2R)))))))))))))
  OK)
```

Hi ha una cosa, però, que no acaba de quadrar. En l'avanç de diagnòstic que fèiem en presentar la incidència que tot just ara acabem de resoldre, havíem dit que, si l'espera no havia anat més enllà dels 6:15 minuts, era "gràcies" al dispositiu de localització de tercets. ¿Havia estat un làpsus o l'afirmació estava fonamentada? Havia estat del tot intencionada, i qui no s'ho cregui pot fer la prova: introduir aquells mateixos 13 valors i activar la casella 8,8, com abans, però utilitzant la versió prèvia a la que detecta configuracions **AAA-BBB** i perfila els tercets a **LLL**; comprovarà com la confirmació de **1** com a candidat encara és més lenta. Així doncs, ¿com s'explica que hi hagi prou a fer aflorar l'estructura implícita dels tercets per reduir dràsticament els temps d'espera en el cas de valors incompatibles, però no en el de valors viables? Doncs perquè inicialment, en exposar els avantatges de tenir els tercets ben predeterminats en **LLL**, potser havíem emfasitzat l'aportació d'intel·ligència sobre la dinàmica cega de **RE-MEMBER** que això suposava (donar-li a conèixer que en una sèrie de tríades només hi havia 3 valors assignables i, per tant, 6 permutacions de 3 elements en comptes de les 504 variacions ternàries de 9 elements, tret de les primàriament incompatibles amb l'entorn), suggerint que era l'escurçament de la regressió zigzaguejant des del punt mort fins a la primera casella sense ocupar la que obrava el miracle, quan no era pas així perquè això hauria suposat una millora insuficient. Tanmateix, la simplificació dels elements de **LLL** corresponents als requadres 9×3 constituïts per tercets repetits tres cops era molt més eficaç: fer possible la intervenció d'**ACTUALITZA-PLUS** (tapant forat amb el tercer valor del tercet, just després que l'usuari n'introduís el segon) o, si la presència de configuracions **AAA-AAA** o **AAA-BBB** afectava dos requadres 9×3, completar la intervenció d'**ACTUALITZA-PLUS** en el primer amb l'acció d'**EXPLORA-3*9** a l'altre per deixar en evidència la infracció de les **normes 2/3** (aquesta vegada únicament en el pla virtual **RE-MEMBER**, amb **INI** = **T**). I, si en el cas de l'última incidència això no funcionava era perquè, amb independència que també participés de la problemàtica dels tercets repetits, el conflicte residia en un altre lloc: també tenia a veure amb infraccions a les **normes 2/3**, però no a nivell requadres 3×3 sinó de files i columnes, i davant del gran cost en temps que hauria suposat ampliar per sistema la cerca **EXPLORA-3*9** a totes aquestes alineacions, i l'escasa probabilitat que certes característiques dels requadres revelessin inequívocament l'existència de línies o columnes infractores i la seva identitat, ha sortit més a compte millorar l'actuació d'**ACTUALITZA-PLUS**, per evitar que s'arribin a produir

aquestes infraccions o per reconduir-les a situacions més fàcilment localitzables per la mateixa funció. Així que no aniria malament tornar a la penúltima versió, i tancar subcapítol i capítol detallant sobre exemples concrets com l'explicitació de l'estructura dels tercets repetits en requadres 9×3 o 3×9 actua resolutivament. Ho farem sobre dos exemples que són gairebé clavats, amb l'unica diferència que en el primer (esquerra) el requadre 9×3 on es repeteixen els tercets **1-2-5**, **3-6-7** i 4-8-9 és l'inferior, mentre que en el segon (centre i dreta) l'hem desplaçat al centre (no hi haurà transposició ni cap reordenació de files ni de requadres 9×3, perquè la primera fila gairebé plena no ho fa necessari). Si en el primer exemple s'han subratllat tant la casella 9,2 com la 5,6 és per indicar que l'exploració a càrrec de **RE-MEMBER** es desenvoluparà igual amb el candidat **5** amb qualsevol de les dues que sigui l'actual. Per la mateixa raó, en el segon exemple s'han subratllat les caselles 9,5 i 5,9 (**5**).

```
0 0 0   0 0 0   0 0 0      0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0

0 0 0   0 5 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 2 0   0 0 0      0 0 0   0 0 0   1 2 5      0 0 0   0 0 0   1 2 5
0 0 0   0 1 0   0 0 0      3 6 7   4 8 1   9 X 0      3 6 7   4 9 8   X 0 0

0 0 0   0 0 0   0 0 0      7 8 9   2 6 5   3 1 4      7 3 2   9 6 1   8 5 4
0 0 0   0 0 0   1 2 5      2 3 6   1 4 9   5 7 8      9 8 6   5 4 2   7 3 1
3 6 7   1 4 2   8 9 X      4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9
```

Pel que fa al primer cas, la representació esquerra mostra l'embranzida inicial en l'exploració del candidat **5** a la casella 5,6 (considerant que 9,2 ja està ocupada) o a la casella 9,2 (considerant que 5,6 ja està ocupada), amb la versió anterior a la preocupació per les configuracions **AAA-BBB**. L'avanç queda interromput a 8,1, per no haver-hi candidats a 9,1, i comencem a recular fins que, després de provar infructuosament el valor 9 a la casella 4,1 sense que hi hagi altres posicions cap on seguir retrocedint, queda confirmada la inviabilitat de **5**. Adoneu-vos que la volada des de l'estat **3-6-7**-1-4-2-8-9-X de la 1ª fila a l'estat **3-6-7**-9-8-5-4-X-0 (no hem cregut que valgués la pena dedicar una segona representació a un canvi tan petit) era molt curta, i per això la no aparició en el caseller 3×3 del candidat **5** no s'ha fet esperar, perquè de seguida han desfilat la resta de valors. Repetir-ho amb la versió que ja té en compte les configuracions **AAA-BBB**, si tenim el candidat **5** a la casella 9,2 (considerant que la 5,6 ja està ocupada), no comportaria canvis sensibles en l'experiència, malgrat que el veredicte de inviabilitat de **5** arribés abans. Però atenció: l'imperceptible guany de velocitat no hauria estat degut a un menor nombre de autorecursions de **RE-MEMBER** pel fet de comprovar només 4, 8 i 9 en el tercer tercet de la 1ª fila, i 1, 2 i 5 en el segon, sinó al fet que la funció **TERCETS** reduiria al valor **(nil nil nil 4 nil nil nil 8 9)** els 3 elements de **R-LLL** corresponents al tercer, i al valor **(1 2 nil nil 5 nil nil nil nil)** només 2 dels 3 elements corresponents al segon, però l'element que correspon a la casella 5,1 es quedaria en **(nil nil nil nil nil nil nil nil nil)** perquè **TERCETS** cauria sobre una llista en què el 1r, 2n i 5è membres són nuls per la presència dels valors **1**, **2** i **5** a la mateixa columna. Però no us penséssiu pas que **RE-MEMBER** hagués d'esperar la primera autorecursió, ocupant virtualment la primera casella lliure (4,1) amb el primer valor 1, per adonar-se de la inviabilitat del candidat **5**, en disparar-se-li l'alarma **OK** a

```
(foreach F (cadr 2R)
  (if (and (not OK) (member (car NIL*NIL) F))
      (setq OK T)))
```

(l'expressió **(car NIL*NIL)** és equivalent a **(nil nil nil nil nil nil nil nil nil)**). La interrupció ja tindria lloc a l'etapa **INI**, per infracció de les **normes 2**/**3** en inspeccionar **EXPLORA-3*9** el segon requadre 3×3 i trobar-se **R-L** amb aquest aspecte

```
(0 0 0 4 0 0 0 8 9)   (0 0 0 4 0 0 0 8 9)   (0 0 0 4 0 0 0 8 9)

(0 0 3 0 0 6 7 0 0)   (0 0 3 0 0 6 7 0 0)   (0 0 3 0 0 6 7 0 0)

(1 2 0 0 5 0 0 0 0)   (0 0 0 0 0 0 0 0 0)   (1 2 0 0 5 0 0 0 0)
```

(com de costum, hem tornat a representar **nil** amb un 0, per raons d'espai), on cal recordar que, encara que la infracció més evident sigui la de la **norma 3** en els elements 1,1 i 3,1 (**1**, **2** i **5**, tres valors només presents en aquestes dues llistes) i **EXPLORA-3*9** es limiti a verificar la **1** i la **3**, sí que registraria la consegüent

infracció d'aquesta última, que en quedar més emmascarada potser demana el concurs d'un ajut: si considerem els altres 7 elements (el 2,1 i els sis superiors) veureu que el conjunt d'aquestes set llistes només conté sis elements: **3**, **4**, **6**, **7**, **8** i **9**. Però si ara considerem que la casella 9,2 ja és plena i deixem els emplenaments en el requadre 3×3 central per al final, tot aniria força diferent: començant per 5,4 (**1**) no notaríem res de particular, però si seguíssim amb 5,5 (**2**) la impossibilitat d'assignar un **5** a la casella 5,6 ja quedaria assegurada pel 5 que **ACTUALITZA-PLUS** hauria col·locat prèviament a 5,1.

Pel que fa al segon, les representacions central i dreta mostren, com hem avancat, l'inici i el fi de l'exploració del candidat **5** a la casella 5,9 (considerant que 9,5 ja està ocupada) o a la casella 9,5 (considerant que 5,9 ja està ocupada), amb la versió anterior a la preocupació per les configuracions **AAA-BBB**. L'embranzida inicial queda interrompuda a 7,4, per no haver-hi candidats a 8,4, i comencem a recular fins que, després de provar infructuosament el valor 9 a la casella 1,2 (no hi ha alternativa al 5 de 2,1 ni al 6 de 8,1), queda palès que **5** és inviable. Adoneu-vos que la volada des de la primera a la segona situació ha estat força més llarga que en el cas precedent, perquè la retirada ha augmentat de 18 caselles i això es tradueix en un temps d'espera inacceptable. A diferència d'aquest cas, ara sí que notarem el progrés en repetir l'experiència amb la versió que té en compte les configuracions **AAA-BBB**, però no ens hem d'enganyar quant a la procedència de l'estalvi, que igual que abans no es deu a la disminució del nombre de retrocessos individuals a la 4ª fila, pràcticament irrellevant, sinó a l'acció de **TERCETS** en deixar en **(nil nil nil nil nil nil nil nil nil)** la llista associada a la casella 5,4, cosa que s'esdevé, com en el cas precedent, per detecció de l'incompliment de les **normes 2/3** en inspeccionar **EXPLORA-3*9** el requadre 3×3 central en la etapa **INI** de **RE-MEMBER**, sense que calgui omplir virtualment la primera casella lliure (4,1) amb el primer valor compatible (5) per veure l'anomalia i provocar la interrupció. Això si el candidat **5** és a la casella 9,5 (considerant que 5,9 ja està ocupada), perquè si, com en el primer cas, suposem que aquesta casella és plena i passem a emplenar les de dalt, tot anirà força diferent: començant per 5,7 (**1**) no hi haurà res de nou però, si emplenem 5,8 (**2**) a continuació, la inviabilitat d'un candidat **5** a la casella 5,9 ja queda assegurada per la presència del 5 que **ACTUALITZA-PLUS** haurà situat prèviament a 5,4. Si finalment repetim els dos casos (que són quatre, si considerem les posicions alternatives del candidat **5**) amb la versió definitiva (la del nou **ACTUALITZA-PLUS**), tot anirà exactament igual.

Aquests dos exemples no s'han triat a l'atzar, sinó per la semblança (sobretot pel que fa al segon) amb el cas que pàgines enrera disparava l'alarma, obligant-nos a explorar línies i columnes per detectar deteminades infraccions de les **normes 2/3**: ens interessava destacar en ambdós casos que, quan **TERCETS** intervé activament en la simplificació del panorama **R-LLL** (a **RE-MEMBER**, quan **INI** = **T**) actua abreujant el nombre i amplitud de les ziga-zagues en l'emplenament virtual de l'escaquer 9×9 i també afavorint que alguns elements d'aquesta pseudomatriu adoptin abans la forma **(nil nil nil nil nil nil nil nil nil)**, però que, si el primer d'aquests efectes és el qui contribueix a escurçar les trajectòries de progressió, el segon és el qui normalment evita les llargues esperes per obtenir el veredicte sobre la idoneïtat d'un valor candidat a la casella actual quan, sigui aquest positiu o negatiu, les trajectòries són regressives. Però també ens donen peu a reobrir la problemàtica de les infraccions a les **normes 2/3** que no es detecten explorant per requadres 3×3 però sí fent-ho per files i columnes, que havia quedat tancada en fals, així que haurem de posposar el desig (expressat a la pàgina precedent) de tancar subcapítol i capítol. De tota manera, ja falta menys...

Si ens centrem en el segon, que és on es percebrien amb més cruesa les tardances si **EXPLORA-3*9** no ens les hagués estalviat, la infracció de les **normes 2/3** en un requadre 3×3 tenia un aspecte molt semblant al mostrat, relatiu al primer exemple (la detecció es produïa en el segon requadre, mentre que ara és el 5è o central).

(0 0 **3** 0 0 **6** **7** 0 0) (0 0 0 0 0 **6** **7** 0 0) (0 0 **3** 0 0 **6** 0 0 0)

(0 0 0 **4** 0 0 0 0 **9**) (0 0 0 **4** 0 0 0 **8** **9**) (0 0 0 **4** 0 0 0 **8** **9**)

(**1** **2** 0 0 **5** 0 0 0 0) (0 0 0 0 0 0 0 0 0) (**1** **2** 0 0 **5** 0 0 0 0)

Com en el primer, tot i que la infracció més evident sigui la de la **norma 3** en els elements 1,1 i 1,3 (**1**, **2** i **5**, tres valors només presents en aquestes dues llistes) i **EXPLORA-3*9** es limiti a verificar la **1** i la **3**, sí que registraria la consegüent

infracció d'aquesta última, que en quedar més emmascarada potser demana el concurs d'un ajut: si considerem els altres 7 elements (el 1,2 i els sis superiors) veureu que el conjunt d'aquestes set llistes només conté sis elements: **3**, **4**, **6**, **7**, **8** i **9**. Però ara volíem anar una mica més enllà, per veure que la infracció també es comet en la 4ª fila: de les 6 llistes que hi té **(cadr 2R)**, els tres valors **1**, **2** i **5** sols apareixen en 2, presents també en el requadre (infracció de la **norma 3**), i en les altres 4 llistes (l'element (0 0 0 0 0 0 0 0 0) i els corresponents a les caselles 6,4, 7,4 i 8,4) únicament apareixen els valors **4**, **8** i **9** (infracció de la **norma 2**). Tant de bo fos sempre així, però hem de recordar que l'**ACTUALITZA-PLUS** que hem fet es limita a solucionar un cas particular d'infracció de les **normes 2**/**3** a nivell de fila i columna: quan dos o més valors només apareixen en una llista de **(cadr 2R)**. Però, ¿què passa quan aquestes infraccions tenen altres característiques, com el cas que acabem de considerar (tres o més valors únicament presents en 2 llistes)? ¿Sempre tindrem la mateixa sort i una infracció detectable sobre la línia també es manifestarà en algun dels tres requadres 3×3 que la contenen? Ben segur que no, perquè aquest exemple era molt especial, en haver-hi una configuració **AAA-BBB** en el 2n requadre 9×3 i donar-se les condicions següents:

A) Les dues úniques llistes de la fila amb **1**, **2** i **5** estaven aliniades i pertanyien al mateix requadre 3×3.

B) En aquestes dues llistes no hi havia més valors que els **1**, **2** i **5** esmentats.

C) Entre les llistes corresponents a les altres alineacions del requadre 3×3 (fins a 6 llistes, corresponents a caselles no ocupades) tampoc no hi havia cap valor **1**, **2** o **5**.

D) La 3ª llista del tercet fila/requadre era buida (formada per 9 membres nuls).

Tot això feia que la infracció de la **norma 2** en el requadre 3×3 es traduís en què hi havia 7 llistes amb només 6 valors. Encara que no s'hagués donat **D** i que a la 3ª llista del tercet hi hagués hagut algun dels valors **3**, **4**, **6**, **7**, **8** i **9**, aquesta infracció s'hauria produït igual. Però si no es compleix **A** i cadascuna de les dues llistes amb **1**, **2** i **5** de la 4ª fila és en un requadre 3×3 diferent, o si deixen de complir-se **B** o **C**, és a dir, els valors **3**, **4**, **6**, **7**, **8** i **9** es barregen amb **1**, **2** i **5** en aquestes llistes i/o algun dels tres últims valors apareix també en els altres tercets del requadre (5ª i 6ª fila), les **normes 2**/**3** tornaran a ser respectades a nivell de requadre 3×3 tot i que s'infringeixin en la 4ª fila sencera: el tàndem format per **EXPLORA-3*9** i **ACTUALITZA-PLUS** encara té vies d'aigua.

0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0
0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0
0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0
0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **5**
0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0
3 6 7	4 8 1	9 X 0	**3 6 7**	0 8 0	9 X 4	**3 6 7**	4 8 1	9 X 0
6 8 9	2 4 5	3 1 7	0 0 0	0 9 0	0 0 0	7 3 6	2 9 5	4 1 8
2 7 3	1 6 9	4 5 8	9 8 2	1 4 6	0 0 0	9 8 2	1 6 4	3 5 7
4 5 **1**	**8 3 7**	**2** 6 **9**	**4** 5 **1**	**8 3 7**	**2** 6 **9**	**4** 5 **1**	**8 3 7**	**2** 6 **9**

0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0
0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0
0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0
0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **5**
0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0
3 6 7	2 4 1	8 9 X	**3 6 7**	4 9 8	X 0 0	**3 6 7**	4 8 1	9 X 0
2 7 6	4 9 5	3 1 8	7 3 2	9 6 1	8 5 4	6 8 9	2 4 5	3 1 7
9 8 3	1 6 2	4 5 7	9 8 6	5 4 2	7 3 1	2 7 3	1 6 9	4 5 8
4 5 **1**	**8 3 7**	**2** 6 **9**	**4** 5 **1**	**8 3 7**	**2** 6 **9**	**4 5 1**	3 7 8	**2 6 9**

No costa massa de trobar un exemple del que dèiem: en l'últim tractat, considereu que la casella actual és 9,5 (**5**) i desplaceu-la un lloc amunt, a 9,6 (1ª renglera, a l'esquerra), amb la qual cosa desfem la configuració **AAA-BBB** que tenia el segon requadre 9×3. Depenent de si utilitzeu la penúltima versió (**ACTUALITZA-PLUS** estès a línies i columnes però sense incorporar encara la detecció a **(cadr 2R)** de valors que només apareixen en una mateixa llista) o l'última (incorporant el dispositiu i abandonant la cerca amb l'activació de **FORA**), l'espera del veredicte negatiu sobre

el candidat 5 trigarà 1 minut i 15 segons o bé sols 8 segons. Per què passa això?: sembla que, si l'última versió resolgués tots els problemes d'incompliment de les **normes 2** i **3** en files i columnes, no hauria de trigar 10 segons, i si no hi pinta res hauria de trigar com l'altra versió (que al seu torn tampoc no ens fa esperar tant com altres vegades). Per comprendre que cada pas donat ajuda a reduir temps (en reduir en nombre i longitud les ziga-zagues del retrocés, sense eliminar-lo) però que no hem aconseguit suprimir d'arrel les exploracions infructuoses (en el sentit que ens hem esforçat a aclarir en les tres pàgines precedents), considereu que la penúltima representació (2ª renglera, al mig) marca la fi de l'exploració, en haver retornat a la primera casella lliure (la 1ª autorecursió de **RE-MEMBER** cal tenir en compte que es produeix amb tota la 1ª fila ocupada, perquè 2,1 (5) i 8,1 (6) s'han emplenat automàticament en omplir 2,4 (**6**)) sense èxit, i les intermèdies (1ª renglera, centre i dreta, i 2ª renglera, esquerra) representen les incursions que precedeixen aquesta, executades segons l'antepenúltima versió, que és la que considerarem de moment. En la inicial veiem que la primera embranzida no ha pogut arribar a 8,4 (X), punt mort des d'on comença la lenta regressió. Quan aquesta ja és a punt d'acabar, amb els darrers cartutxos a 1,2 (9) i 2,2 (8), veurem com amb un 2 a 3,2 es van alternant dues situacions. Quan el segon tercet de la 2ª fila és 1-4-6 (1ª renglera, centre), els valors 4 i 6 provoquen un seguit d'emplenaments automàtics que enumerarem seguint l'ordre en què es produeixen: a 5,3 (9), a 5,4 (8), a 9,4 (4) i a 7,4 (9), després dels quals la casella 8,4 quedarà en punt mort (a **(cadr 2R)**, la llista corresponent serà **(nil nil nil nil nil nil nil nil nil)**) i això avortarà la incursió. El mateix passarà quan el tercet esmentat sigui 5-4-6 i 6-4-0 (el zero significa que n'hi haurà prou a emplenar virtualment les caselles 4,2 i 5,2 perquè es desencadeni la mateixa seqüència d'emplenaments automàtics). Quan aquest tercet sigui 1-6-4 (1ª renglera, dreta), 4-6-1, 4-6-5 i 5-6-4, no hi haurà tanta sort, perquè només s'esdevindrà un emplenament automàtic (a 5,3 (9), amb 1-6-4 i 5-6-4) o dos (el propi valor 6 del tercet, després d'haver introduït el 4, amb 4-6-1 i 4-6-5), a més dels habituals de final de fila i de requadre 3×3. En conjunt, però, s'haurà produït cert abreujament de les giragonses en relació al temps que hauria durat l'exploració sense aquestes incidències. (Ens hem limitat a les tríades que tenen possibilitats de completar la fila en no entrar en conflicte amb el tercer tercet de la 1ª fila.)

Amb un 3 a 3,2 s'alternen tres situacions. Quan el segon tercet de la 2ª fila és 1-6-2 (2ª renglera, esquerra), 2-6-1 i 5-6-2, a l'embranzida inicial li falta poc per completar la 4ª fila i tindrem un retrocés zigzaguejant fins aquesta posició. Quan el tercet és 2-4-6 i 6-4-0 es repeteix la seqüència d'emplenaments automàtics que hem descrit en l'últim paràgraf i que conclou amb una casella en punt mort que posa fi a cada incursió. I quan sigui 2-6-4 i 4-6-2 tindrem un i dos emplenaments automàtics respectivament, a més dels habituals, com els que que també hem descrit en el paràgraf precedent. (Com allà, ens hem limitat a les tríades que no entren en conflicte amb el tercer tercet de la 1ª fila.)

Ja per acabar, amb un 6 a 3,2 cloem l'exploració en retornar sense èxit a 1,2 (9). De les quatre tríades de valors amb possibilitats de completar la 2ª fila (seguim parlant del segon tercet), 1-4-2, 2-4-1, 2-4-5 i 5-4-2, hem dit que representàvem l'última (2ª renglera, al mig), i només destacarem que el valor central 4 de totes elles s'assigna automàticament (a 5,2) només emplenar 3,2 (6), circumstància que haurà contribuït a alleugerir l'espera, amb els altres forats tapats habitualment (final de fila i de requadre 3×3). No ens hem molestat a mesurar el temps, però és clar que queda molt per sobre del enregistrat per la penúltima versió (1 minut i 15 segons) que considerarem tot seguit, tot i que estudiant-ne el comportamentent en les mateixes 5 situacions ningú no hauria pronosticat un progrés considerable.

La representació inicial (de nou, 1ª renglera, a l'esquerra), igual que les altres quatre, no mostra que la incursió hagi estat més ràpida (fons i tot sembla just el contrari, perquè el cul-de-sac 8,4 es desplaça una casella endavant), però això és degut a la poca expressivitat d'aquestes representacions (que tampoc no hem volgut complicar amb més subratllats, parèntesis o altres diferenciacions tipogràfiques que podrien confondre abans que aclarir): per seguir el rastre de les ziga-zagues no hi ha més remei que llegir atentament aquestes línies, tot i que ara no tindreu la incomoditat de passar plana per veure els fotogrames. Per no ser reiteratius, direm una vegada per totes que l'escurçament de trajectes es deu al fet que alguns emplenaments són automàtics (més que abans, perquè la diferència fonamental entre l'antepenúltima versió i la penúltima que considerem ara és **ACTUALITZA-PLUS**, que ha ampliat l'àmbit d'actuació) i això comporta que algunes ocupacions de caselles s'ajunten en una mateixa execució d'**ACTUALITZA** (com quan un vol treure una cirera

del cistell i se n'enduu un grapat sense voler), de manera que entrades i sortides de l'autorecursió de **RE-MEMBER** (avanços i retrocessos en l'escaquer 9×9) afecten a blocs de caselles i no a una de sola. En concret, en aquesta primera representació l'emplenament de 4,2 (1) arrossega el de 6,4 (1), el de 4,3 (2) arrossega el de 4,4 (5) i el de 5,4 (8) arrossega el de 8,4 (4) i el de 7,4 (8), però igual passa a la tercera (1ª renglera, dreta). A la segona (1ª renglera, centre), es dóna la mateixa reacció en cadena que en la versió precedent (5,3 (9), 5,4 (8), 9,4 (4) i 7,4 (9), rematada pel cul-de-sac 8,4) però prèviament, després d'emplenar 4,2 (1), es tapa el forat 6,4 (1), com a la primera i tercera representació. Aquest mateix forat tapat és l'únic que podem afegir, en la quarta representació (2ª renglera, esquerra), al que havíem descrit per a la versió precedent. I pel que respecta a l'embranzida final (2ª renglera, centre), com en aquesta tenim que l'emplenament de 3,2 (6) desencadena el de 5,2 (4) però ara, a més, el de 4,2 (5) desencadenarà el de 6,4 (5), circumstància que donarà lloc a que, després d'emplenar 5,4 (9), també s'empleni la casella 8,4 (8), més enllà del cul-de-sac 7,4.

```
0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0
0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0
0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0

0 0 0   0 0 0   0 0 5      0 0 0   0 0 0   0 0 5      0 0 0   0 0 0   0 0 5
0 0 0   0 0 0   1 2 0      0 0 0   0 0 0   1 2 0      0 0 0   0 0 0   1 2 0
3 6 7   5 8 1   9 4 X      3 6 7   0 8 1   9 X 4      3 6 7   5 8 1   9 4 X

6 8 9   2 4 5   3 1 7      0 0 0   0 9 0   0 0 0      7 3 6   2 9 5   4 1 8
2 7 3   1 6 9   4 5 8      9 8 2   1 4 6   0 0 0      9 8 2   1 6 4   3 5 7
4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9
```

```
0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0
0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0
0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0

0 0 0   0 0 0   0 0 5      0 0 0   0 0 0   0 0 5      0 0 0   0 0 0   0 0 5
0 0 0   0 0 0   1 2 0      0 0 0   0 0 0   1 2 0      0 0 0   0 0 0   1 2 0
3 6 7   2 4 1   8 9 X      3 6 7   4 9 5   X 8 0      3 6 7   5 8 1   9 4 X

2 7 6   4 9 5   3 1 8      7 3 2   9 6 1   8 5 4      6 8 9   2 4 5   3 1 7
9 8 3   1 6 2   4 5 7      9 8 6   5 4 2   7 3 1      2 7 3   1 6 9   4 5 8
4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9      4 5 1   3 7 8   2 6 9
```

Són més evidents els mecanismes que justifiquen el considerable guany de temps de l'última versió en relació a la penúltima: a cap de les 5 situacions representades tot seguit l'embranzida no necessita anar més enllà de 4,2 per veure que no hi ha futur, perquè aquest emplenament desencadena el de 6,4 com en la versió precedent, i això activa el dispositiu de detecció d'una llista **E-LL3** portadora de dos valors exclusius en la 4ª fila i que correspon a la casella 4,4 (X), detecció que avorta l'exploració per al candidat **5** de 9,6; l'única diferència entre les 4 primeres i la cinquena (2ª renglera, centre) és el valor assignat virtualment a 4,2 de primer i a 6,4 de retruc (1 i 5, respectivament) i el parell de valors que deteminen la singularitat de la llista **E-LL3** detectada (2 i 5, i 1 i 2). Amb uns escurçaments tan dràstics no té res d'estrany que el temps d'espera s'hagi reduït a la novena part (de 75 a 8 segons):

```
0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0      0 0 0   0 5 0   0 0 0
0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0      0 0 0   0 2 0   0 0 0
0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0      0 0 0   0 1 0   0 0 0

0 0 0   0 0 0   0 0 5      0 0 0   0 0 0   0 0 5      0 0 0   0 0 0   0 0 5
0 0 0   0 0 0   1 2 0      0 0 0   0 0 0   1 2 0      0 0 0   0 0 0   1 2 0
3 6 7   X 0 1   0 0 0      3 6 7   X 0 1   0 0 0      3 6 7   X 0 1   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
2 7 3   1 0 0   0 0 0      9 8 2   1 0 0   0 0 0      9 8 2   1 0 0   0 0 0
4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9      4 5 1   8 3 7   2 6 9
```

0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0
0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0
0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0
0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **5**
0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0
3 6 7	X 0 1	0 0 0	**3 6 7**	X 0 5	0 0 0	**3 6 7**	X 0 1	0 0 0
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
9 8 3	1 0 0	0 0 0	9 8 6	5 0 0	0 0 0	2 7 3	1 0 0	0 0 0
4 5 **1**	**8 3 7**	**2** 6 **9**	**4** 5 **1**	**8 3 7**	**2** 6 **9**	**4 5 1**	3 7 8	**2 6 9**

Aquesta durada (8 segons) no és que sigui admissible, ni de bon tros, però ens pot fer vacil·lar en el sentit que, si bé és prou clar que el conflicte no ha quedat solventat d'arrel, detectant que els valors **1**, **2** i **5** només apareixen dos cops a la 4ª fila de la pseudomatriu **(cadr 2R)** (en les llistes corresponents a les posicions 4,4 i 6,4) així que el valor **5** s'assigna virtualment a la casella 9,6, la detecció d'una llista única amb **2** i **5** en un cas, i amb **1** i **5** a l'altre, hi ha ajudat prou. La temptació d'acceptar aquests 8 segons com a mal menor a canvi de deixar-ho tot com estava era massa forta, i per això ha convingut una última maniobra per cremar les naus com l'Hernán Cortés: és dubtós que 8 segons d'espera per a un únic valor candidat siguin acceptables, però tohom estarà d'acord que 50 segons ja no ho són. Així que ja podem preparar-nos per ampliar l'acció dissuasiva d'**ACTUALITZA-PLUS** al supòsit presentat, perquè costa ben poc multiplicar l'espera per 6 amb un senzill artifici: buidar les caselles 4,1, 5,1 i 6,1, emplenant 2,1 (**5**) i 8,1 (**6**) en justa compensació (així queda garantit que no es permutaran automàticament els requadres 9×3), de forma que el veredicte d'incompatibilitat de **5** a 9,5 no arribarà fins que l'exploració no hagi esgotat les 6 permutacions (3-7-8, 3-8-7, 7-3-8, 7-8-3, 8-3-7 i 8-7-3) del segon tercet. ¿Endevineu ara per què en cadascun dels tres blocs de 6 representacions només n'havíem utilitzat les 5 primeres?: a l'última representació de cada bloc teniu l'embranzida inicial en la versió corresponent (antepenúltima, penúltima i última).

En unes altres circumstàncies i vist que, a més de contemplar l'activació de **FORA** en **ACTUALITZA-PLUS** quan 2 o més valors apareguin només en 1 llista, també ho hem de fer quan 3 o més valors apareguin només en 2 llistes, no ens ho pensaríem més i generalitzaríem el procediment, com ho havíem fet a **EXPL-SUBCONJ** < **EXPLORA-3*9**, però, havent acumulat una densitat d'operacions considerable, creiem que no seria seriós tirar pel dret sense haver-nos assegurat que no hi ha més remei que fer-ho. Tal vegada d'aquí a uns pocs anys, quan la velocitat dels processadors estàndard s'hagi triplicat, molts dels tractaments particulars muntats per evitar que totes les línies convergeixin en un procés únic, senzill però costós en termes de temps, deixaran de tenir sentit, i serà més pràctic ignorar les especificitats i anar cap a la solució universal. Però ara, no sabent si tirar pel dret és assumible, sembla raonable insistir una mica més. Així, després d'haver esmerçat infructuosament una quantitat considerable de temps a intentar demostrar per la via deductiva que si, entre les $\mathbf{9-k}$ llistes d'una fila o columna de **(cadr 2R)** amb **k** caselles ocupades hi havia $\mathbf{n>3}$ valors que sols apareixien a **m** llistes ($\mathbf{m<n}$) o que les restants $\mathbf{9-(k+m)}$ llistes només tenien $\mathbf{9-(k+n)}$ valors (que, si es compleix la **norma 1**, són formulacions equivalents), també havia de passar alguna cosa semblant en els requadres 3×3 (amb $\mathbf{m<n}$, encara que **k**, **m** i **n** ja no fossin els mateixos valors), l'autor va decidir (potser carregat d'excessiva bona fe i a risc de ser objecte de mofa i befa per acabar fent els comptes de la vella) provar de sortir-se'n cercant contraexemples que avalessin la tesi contrària. A més dels dos casos que coneixem, n'ha pogut trobar tres: de 4 valors que només apareixen a tres llistes, de 5 que sols apareixen a quatre i de 6 que només apareixen a cinc. Veieu-los representats:

0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 **6**	0 0 0	0 0 0
0 0 0	0 0 0	0 **4 5**	0 0 **5**	0 0 0	**4** 0 0	0 0 **5**	0 0 0	0 0 0
5 4 0	0 0 0	0 0 0	0 0 **4**	0 0 0	**5** 0 0	0 0 **4**	0 0 0	0 0 0
0 0 0	0 0 0	0 **2 1**	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 **1 2**	0 0 0	0 0 0	0 0 0	**4** 0 **5**	0 0 0	0 0 0	**4 5 6**	0 0 0
0 0 **3**	0 0 0	**6** 0 0	0 0 0	**1 2 3**	0 0 0	0 0 0	**1 2 3**	0 0 0
0 0 0	0 0 0	0 0 0	0 0 **3**	0 0 0	**1** 0 0	0 0 **3**	0 0 0	**9** 0 0
0 0 0	0 0 0	0 **5 4**	0 0 **2**	0 0 0	**3** 0 0	0 0 **2**	0 0 0	**8** 0 0
4 5 0	**8 3 7**	0 0 0	0 0 **1**	0 0 0	**2** 0 0	0 0 **1**	0 0 0	**7** 0 0

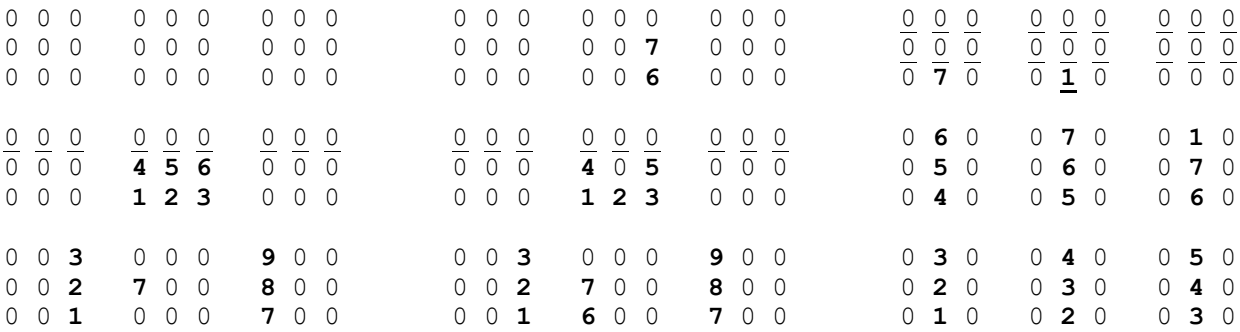
Com que, per mostrar que les **normes 2**/**3** es compleixen escrupolosament a nivell de requadre 3×3, en cada cas hauríem de reproduir almenys els tres fragments **R-L** que corresponen als requadres afectats per la fila conflictiva i això resultaria massa farragós, ens limitarem a mostrar aquesta fila de **(cadr 2R)** i, per raons d'espai (encabir-la en una única línia de text obligaria a comprimir massa els caràcters), la presentarem de manera simplificada però per partida doble: de primer, i per fer palès l'incompliment de la **norma 3**, els elements de la fila aliens a les llistes sobrecarregades els representarem amb **X** (tant si són **nil** com si també són llistes) i d'aquestes només explicitarem els valors que excedeixen en nombre al de llistes, separats per punts suspensius si no són correlatius; en segon lloc, per fer palès l'incompliment de la **norma 2**, els elements aliens a les llistes infracarregades els representarem amb **X** (tant si són **nil** com si també són llistes) i d'aquestes només explicitarem els valors no nuls, separant-los per punts suspensius si no són correlatius; si, malgrat la simplificació, la representació de la fila no capigués en una línea de text (llistes amb molts valors significatius i/o molt nombroses), en lloc d'explicitar **m** llistes només n'hi haurà una, seguida de l'expressió "×m". En segon lloc (entre parèntesis) hi ha instruccions sobre l'ordre d'emplenament:

- Pàg. precedent, a l'esquerra (acabeu amb 3,4)- 4ª fila (2 caselles ocupades):
  - 4 valors només presents a 3 llistes (infracció de la **norma 3**):
    X X X (**1 2 ... 4 5** ...) (**1 2 ... 4 5** ...) (**1 2 ... 4 5** ...) **X X X**
  - 4 llistes amb només 3 valors (infracció de la **norma 2**):
    (... **7 8 9**) (... **7 8 9**) **X X X X X** (... **7 8 9**) (... **7 8 9**)
- Pàg. precedent, al mig (l'ordre és indiferent).- 6ª fila (cap casella ocupada):
  - 5 valors només presents a 4 llistes (infracció de la **norma 3**):
    (**1 2 3 4 5** ...) (**1 2 3 4 5** ...) **X X X X X** (**1 2 3 4 5** ...) (**1 2 3 4 5** ...)
  - 5 llistes amb només 4 valors (infracció de la **norma 2**):
    X X (... **6 7 8 9**) (... **6 7 8 9**) (... **6 7 8 9**) (... **6 7 8 9**) (... **6 7 8 9**) **X X**
- Pàg. precedent, a la dreta (comenceu amb **7**-**8**-**9**).- 6ª fila (cap casella ocupada):
  - 6 valors només presents a 5 llistes (infracció de la **norma 3**):
    (**1 2 3 4 5 6** ...) (**1 2 3 4 5 6** ...) **X X X X** (**1 2 3 4 5 6** ...)×3
  - 4 llistes amb només 3 valors (infracció de la **norma 2**):
    X X (... **7 8 9**) (... **7 8 9**) (... **7 8 9**) (... **7 8 9**) **X X X**

A la vista del resultat, si no hagués estat per l'heterogeneïtat dels tres casos, hauríem acabat abans i probablement hagués estat més entenedor limitar-se a posar:
- Esquerra (4ª fila): **1**, **2**, **4** i **5** només admissibles en 4ª, 5ª i 6ª posició.
- Centre (6ª fila): **1**, **2**, **3**, **4** i **5** només admissibles en 1ª, 2ª, 8ª i 9ª posició.
- Dreta (6ª fila): **1**, **2**, **3**, **4**, **5** i **6** només admissibles en 1ª, 2ª, 7ª, 8ª i 9ª pos.

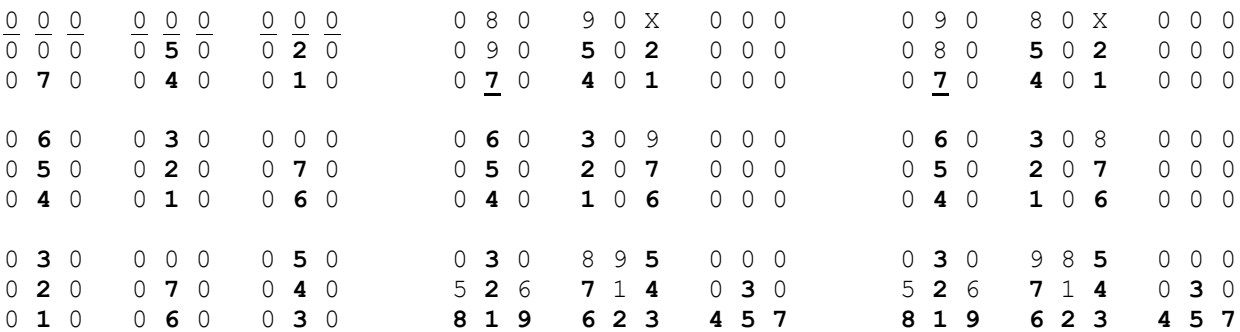
Ho tindrem present d'ara endavant, per si s'escau donar alguna informació similar.

Força més ha costat de trobar un cas on una fila de **(cadr 2R)** tingués 7 valors que només apareguessin a sis llistes, sense que aquesta infracció de les **normes 2**/**3** es manifestés en algun requadre 3×3. Com que els primers intents realitzats sense cap orientació concreta (només que mantenint la mirada posada a la 6ª fila, com en els dos últims casos, i pensant amb els valors **1**, **2**... **7**) resultaven estèrils, l'autor va pensar en seguir alguna pauta, ni que fos per assegurar l'exhaustivitat de la cerca si finalment tot apuntava a la conclusió que el cas esmentat era imposible, i va creure que un bon criteri de classificació podria ser la localització de les 6 llistes amb els 7 valors. Així, considerant una distribució per requadres 3×9, que serveix per a qualsevol fila, pot ser que: 1) tres d'aquestes llistes n'ocupin un, les altres tres n'ocupin un altre i en el tercer requadre ja no n'hi hagi cap; 2) a cadascun dels requadres hi hagi dues d'aquestes llistes; 3) a un requadre hi hagi tres d'aquestes llistes, a l'altre n'hi hagi dues i en el tercer només n'hi hagi una. La inviabilitat del primer cas es fa palesa a la representació esquerra, on hem modificat algunes coses de l'últim exemple per tal que en el tercet central de la 6ª fila cap valor **1**, **2**... **6** no sigui admissible, però a partir d'aquí no hi ha manera d'aconseguir que tampoc no ho sigui el **7** i que, tanmateix, aquest valor sigui admissible en alguna casella del primer i tercer tercet: la seva presència és obligada en el tercet central de la 6ª fila i això fa que estigui vetat en els altres dos. Com a molt, situant un **7** en el requadre 3×3 de sota aconseguirem que el valor deixi de ser admissible en una casella d'aquest tercet central, perquè posant-ne un segon en el requadre 3×3 superior central l'espifiaríem del tot, en tapar **ACTUALITZA-PLUS** amb un altre **7** el forat obert en el tercet. La representació següent mostra una millor aproximació al nostre propòsit: com abans, aconseguim que **1**, **2**... **7** siguin valors admissibles en la major part de caselles del primer i tercer tercet de la 6ª fila, però no hi ha hagut forma d'evitar que els reticents **6** i **7** també ho siguin en una casella del centre. El lector ja pot fer totes les provatures que vulgui, però tampoc no se'n sortirà. Passem, doncs, al segon cas.



Així com en el primer la inviabilitat es dirimia en el requadre 3×9 central, en el segon cas, representat a la dreta, cal mirar el conjunt dels tres requadres 3×9, on a cada columna central volíem situar els valors **1**, **2**... **7**, deixant lliures les dues files superiors de manera que les caselles 2,7, 5,7 i 8,7 quedessin ocupades pels valors **7**, **1** i **2**, respectivament, i fos a la 8ª fila (i a la 9ª, de retruc) on es plantegés la infracció de les **normes 2**/**3**: que únicament en 6 llistes **(cadr 2R)** (1ª i 3ª de cada tercet) apareguessin els valors **1**, **2**... **7**, mentre que les llistes corresponents a les caselles 2,8, 5,8 i 8,8 només tinguessin valors **8** i **9**. Aquesta era la intenció que, ultra no haver-se pogut materialitzar, com el cas precedent, ha comportat trobar-se prematurament a l'espera d'un veredicte (negatiu) d'aquells que no arriben mai, i diem prematurament perquè la incidència s'ha presentat abans d'hora, quan no hi havia cap infracció que la justifiqués. Per poder identificar més clarament la causa de l'imprevist (no pas ara, sinó en el subcapítol següent), recomanem completar els emplenaments de les sis primeres files abans de passar a 2,7 (**7**) i 5,7 (**1**, subratllat per destacar-lo com l'últim emplenament i com a valor candidat que motiva la llarga pausa). De moment, però, sí que cridem l'atenció del lector sobre el fet que aquest límit, que no hem pogut ultrapassar per arribar al supòsit de 7 valors només presents a 6 llistes, no comporta cap infracció de les **normes 2**/**3** a nivell de la 7ª, 8ª o 9ª fila ni dels tres requadres 3×3 superiors, circumstància que ja ens du a pronosticar que la versió que estendrà l'exploració a files i columnes seguirà estavellant-se contra casos semblants (és per això que hem parlat d'un nou subcapítol). També ara ens podem permetre explicar per què el candidat **1** és desestimat a 5,7 i la demora d'aquest veredicte d'incompatibilitat: després de l'emplenament 2,7 (**7**), les caselles 2,8 i 2,9 només admetran els valors **8** i **9**; després d'un emplenament 5,7 (**1**), les caselles 5,8 i 5,9 només admetrien els valors **8** i **9**; la conseqüència d'aquesta hipòtesi és que 8,8 i 8,9 sols podrien admetre el **2** (les dues alhora?) mentre que 8,7 podria admetre els valors **2**, **8** i **9** (hauria de ser ocupada a la vegada pel **7** i pel **8**?), ambdues coses un despropòsit. Però el perfil contradictori no queda reflectit en les llistes **(cadr 2R)**, i aquest és el problema: una infracció latent de les **normes 2**/**3**, a nivell de la 8ª columna (2 valors en només 1 llista i 2 llistes amb només 1 valor), que no serà detectada.

Seguint amb la nostra cerca d'algun exemple de fila amb 7 valors només presents en 6 llistes, per fi el trobarem en la mateixa segona modalitat: només cal modificar el cas precedent de manera que tinguem lliure la 9ª fila (serà la de referència) i que, sobre les columnes ocupades (2ª, 5ª i 8ª), cadascuna de les altres caselles lliures se situï en un requadre 9×3 diferent (això obligarà a desplaçar la columna del mig, per evitar coincidències). El teniu a l'esquerra i l'ordre d'emplenament pot ser qualsevol. Serà en activar l'última casella que veureu com el candidat que correspongui triga a "no sortir" en el caseller 3×3: després d'una pausa d'uns dos minuts, apareixeran els únics candidats acceptats, **8** i **9**. A diferència de l'atípic exemple precedent, amb la versió que explorarà files i columnes no caldrà esperar.



Havent provat la possibilitat que en una fila hi hagi 7 valors en només 6 llistes (dels valors **1**, **2**... **7**, la 1ª i la 3ª tenen els valors **1**, **2**, **3**, **4**, **5** i **6**; la 4ª i la 6ª, els valors **1**, **2**, **3**, **6** i **7**; la 7ª i la 9ª, els valors **3**, **4**, **5**, **6** i **7**) perquè les altres 3 llistes (la 2ª, 5ª i 8ª) només tenen 2 valors (**8** i **9**), sense que les **normes 2**/**3** s'incompleixin en cap requadre 3×3 (no ens entretindrem comprovant-ho), no caldria insistir-hi més, però com que abans hem parlat d'una tercera modalitat (que en un tercet hi hagi tres d'aquestes llistes, a l'altre n'hi hagi dues i en el tercer només n'hi hagi una) mirarem si això també és possible en aquesta forma. En un únic exemple, que hem desdoblat en les representacions central i dreta i que a més recorda molt l'anterior (observeu que les columnes en negreta tenen idèntica composició, però que s'han corregut horitzontalment), en tindrem prou per copsar com sovint és difícil retenir una determinada configuració, per la volatilitat que li dóna la intervenció d'**ACTUALITZA-PLUS** tapant forats. De primer, ens disculparem per la incomoditat que comporta llegir en aquesta plana la descripció dels fets i haver de tornar a la precedent per situar-los en les representacions esmentades, en què els valors en negreta corresponen a les caselles emplenades realment, el **7** de 2,7 (casella que cal activar al final) correspon al candidat refusat (**8** i **9** són admesos) i els caràcters sense ennegrir corresponen als emplenaments virtuals per avaluar aquesta candidatura. Perquè, en principi, la 9ª fila havía de ser un altre exemple de fila amb 7 valors en només 6 llistes (de fet, ho és), però l'actuació d'**ACTUALITZA-PLUS** fa innecessària la nova versió que explorarà files i columnes: amb l'actual, el veredicte negatiu sobre l'aptitud de **7** arribarà sense cap demora. Si considerem les tres caselles lliures del requadre 3×3 inferior-centre, veurem que 5,2 i 5,3 admeten els valors 1, 8 i 9, mentre que 4,3 només admet 8 i 9. Doncs bé: a la representació del mig hi tenim el valor 8 (amb 1 a 5,2 i 9 a 5,3, però si aquests valors estiguessin permutats passaria exactament el mateix), i a la dreta hi tenim el 9 (amb 1 a 5,2 i 8 a 5,3, però si aquests valors estiguessin permutats passaria exactament el mateix); com a emplenaments virtuals previs només tenim el 5 a 1,2 i el 6 a 3,2, o bé aquests valors permutats; en total seran, doncs, quatre brevíssimes incursions com la del centre i quatre més com la de la dreta. Hem dit brevíssimes perquè no caldrà emplenar tots els buits fins al cul-de-sac situat a la casella 6,9 (X), sinó limitar-se als representats, la major part dels quals són automàtics: en el centre, després dels emplenaments virtuals ordinaris 5,2 (1) i 4,3 (8) (quan sigui 5,2 (8), l'emplenament 4,3 (9) ja serà automàtic), vindrà la seqüència d'emplenaments automàtics 5,3 (9), 4,9 (9), 2,9 (8), 2,8 (9) i 6,6 (9), després del qual 6,9 es queda sense valors admissibles; a la dreta, després dels emplenaments virtuals ordinaris 5,2 (1) i 4,3 (9) (quan sigui 5,2 (9), 4,3 (8) ja serà automàtic), vindrà la seqüència d'emplenaments automàtics 5,3 (8), 4,9 (8), 2,9 (9), 2,8 (8) i 6,6 (8) després del qual 6,9 es queda sense valors admissibles. Les vuit incursions-llampec (gairebé sense retrocés, perquè majoritàriament estan compostes d'emplenaments automàtics) justifiquen que el veredicte sigui tan ràpid.

I ja ho podem deixar aquí, perquè ordenant adequadament la detecció i ocupació de forats per tapar, i la detecció d'elements **(nil nil nil nil nil nil nil nil nil)** a **(cadr 2R)** (que, per fer-ho més curt, de vegades representarem amb l'equivalència **(car NIL*NIL)** i que cal no confondre amb els elements **nil** corresponents a caselles ocupades) i d'infraccions a les **normes 1** i **2**/**3**, d'aquestes darreres n'hi ha prou a considerar els 6 tipus que sabem que ens podem trobar perquè n'hem vist exemples. No cal pensar en files, columnes o requadres 3×3 amb 8 valors que només apareguin a set llistes o 9 que només apareguin a vuit, perquè la prèvia detecció d'elements **(car NIL*NIL)** haurà causat l'abandó de l'exploració i el veredicte d'inviabilitat del candidat que l'havia posada en marxa. L'última hipòtesi comporta que la novena llista és **(car NIL*NIL)** (cal que sigui una llista plena, i no pas **nil**, perquè amb **k** caselles ocupades -**k** = **0**, **1**... **9**- només podem parlar de **9 − k** valors assignables i de **9 − k** llistes no nul·les) i haurà estat detectada. Pel que fa a la penúltima, que 8 valors només apareguin a set llistes vol dir, en el millor dels casos, que les altres dues llistes sols tenen un valor no nul (el nové), i llavors es posarà en marxa **ACTUALITZA_PLUS** per tapar el primer forat localitzat; amb una casella ja ocupada per aquest nové valor, la llista **(cadr 2R)** que correspon l'altra quedarà reduïda a **(car NIL*NIL)**, serà detectada i, com abans, l'exploració es cancel·larà. I, per descomptat, igual com hem afirmat que podíem ignorar l'existència d'unitats amb 8 valors que només apareguessin a set llistes o amb 9 que només apareguessin a vuit, aplicant-ho a files, columnes i requadres 3×3 sense fer distincions, perquè l'argumentació valia per als tres supòsits, ara no cometrem la ingenuïtat de fer com amb les llistes i posar-nos a buscar exemples de requadres 3×3 representatius de tota la gamma (ja no parlem de columnes, perquè és obvi que els mateixos casos servirien d'exemple, transposant l'escaquer 9×9 i canviant el farciment per evitar que la transposició automàtica anul·lés la manual), per diverses raons: perquè ja

en devem arrossegar molts fins ara, d'exemples d'aquests; per una molt justificada mandra... però, sobretot, perquè el reguitzell de 6 exemples que ens hem esforçat a reunir potser pretenia ser exhaustiu en el moment de posar-nos-hi, però després de tant joc malabar hem tingut ocasions per adonar-nos que no ho era en absolut. En efecte, si buscàvem criteris taxonòmics sense ambigüitat, no els hem sabut usar correctament, perquè dos subcapítols enrera ja havíem advertit que les **normes 2** i **3** només eren equivalents si es compleix la **norma 1**, i ara caldria afegir que, per quantificar aquesta dualitat, hem de quantificar els dos sumands que determinen la **norma 1**. Clar i català: hem de partir del nombre de caselles ocupades en la unitat (fila, columna o requadre 3×3) perquè si ens limitem a indicar com s'incompleix la **norma 3** o com s'incompleix la **norma 2** el cas no queda inequívocament identificat. Sense ser-ne del tot conscients, a l'hora de descriure els incompliments teníem al cap unitats amb totes 9 caselles lliures però no paràvem esment al conjunt de les tres variables, i així anàvem treient-nos de la màniga uns exemples que no sempre corresponien a aquesta presumpció. Volíem demostrar que, en files i columnes, era possible que 2 o més valors només apareguessin en 1 llista (i que en 8 llistes només apareguessin 7 o menys valors), 3 o més valors només apareguessin en 2 llistes (i en 7 llistes només apareguessin 6 o menys valors), 4 o més valors només apareguessin a 3 llistes (i en 6 llistes només apareguessin 5 o menys valors), 5 o més valors només apareguessin a 4 llistes (i en 5 llistes només apareguessin 4 o menys valors), 6 o més valors només apareguessin a 5 llistes (i en 4 llistes només apareguessin 3 o menys valors) i que 7 o més valors només apareguessin a 6 llistes (i que en 3 llistes només apareguessin 2 o menys valors). Però aquesta relació no era completa ni corresponia ben bé als casos que realment contemplàvem, que eren:

- Fila amb 3 caselles ocupades (amb els valors **5**, **8** i **9**):
  - 2 valors (**1** i **2**) només en 1 llista.
  - 5 llistes amb només 4 valors (**3**, **4**, **6** i **7**).
- Fila amb 3 caselles ocupades (amb els valors **3**, **6** i **7**):
  - 3 valors (**1**, **2** i **5**) només en 2 llistes.
  - 4 llistes amb només 3 valors (**4**, **8** i **9**).
- Fila amb 2 caselles ocupades (amb els valors **3** i **6**):
  - 4 valors (**1**, **2**, **4** i **5**) només en 3 llistes.
  - 4 llistes amb només 3 valors (**7**, **8** i **9**).
- Fila amb 0 caselles ocupades:
  - 5 valors (**1**, **2**, **3**, **4** i **5**) només en 4 llistes.
  - 5 llistes amb només 4 valors (**6**, **7**, **8** i **9**).
- Fila amb 0 caselles ocupades:
  - 6 valors (**1**, **2**, **3**, **4**, **5** i **6**) només en 5 llistes.
  - 4 llistes amb només 3 valors (**7**, **8** i **9**).
- Fila amb 0 caselles ocupades:
  - 7 valors (**1**, **2**, **3**, **4**, **5**, **6** i **7**) només en 6 llistes.
  - 3 llistes amb només 2 valors (**8** i **9**).

Per què diem que no és completa?: perquè només els tres últims exemples tenen tota la fila lliure. ¿On són els altres de fila lliure amb 2 valors només en 1 llista, 3 valors només en 2 llistes i 4 valors només en 3 llistes? ¿On són els de fila amb 1 casella ocupada i ...? ¿On són els altres de fila amb 2 caselles ocupades i 2 valors només en 1 llista, 3 valors només en 2 llistes, 5 valors només en 4 llistes i 6 valors només en 5 llistes? ¿On són els altres de fila amb 3 caselles ocupades i 4 valors només en 2 llistes, 5 valors només en 4 llistes i 6 valors només en 5 llistes? ¿On són els de fila amb 4 caselles ocupades i ...? ¿On són els de fila amb 5 caselles ocupades i ...? ¿On són els de fila amb 6 caselles ocupades i ...?. La relació completa la teniu cinc pàgines endavant (pàg. 308 i 309), i s'ha muntat aplegant les infraccions teòriques (no totes les llistades es donen a la pràctica) segons el nombre de caselles ocupades de la unitat (fila, columna o requadre 3×3) i prioritzant la formulació relativa a la **norma 3** (que figura en primer lloc, ja que la pretesament exhaustiva sistemàtica de la majoria d'exemples vistos atenia sobretot a aquest aspecte) sobre la relativa a la **norma 2** (que és en segon lloc). No les hem portat aquí perquè consideràvem que ja era hora de coneixer l'impacte que tindria l'ampliació a files i columnes d'una exploració que fins ara afectava únicament els requadres 3×3 i, un cop apamat això, ja veuríem si aquestes llistes admetien alguna retallada. Trobar exemples que no cobreixen tota la tipologia però que són prou variats i acceptablement representatius, haurà servit almenys perquè ens adonem que, si la primera hipòtesi adoptada (només en requadres 3×3 ...) era falsa, la segona (... tret de rares excepcions en files i columnes) també ho era: les infraccions de les **normes 2**/**3** en llistes o columnes no pressuposen que sempre hagin de quedar reflectides en requadres 3×3, ni a l'inrevés, i en tots tres llocs poden respondre a qualsevol de les formes representades en aquelles dues pàgines.

I, ja posats a pensar en com reduir l'impacte de què parlàvem, ens sap greu haver de desil·lusionar aquells que se'n guardaven una d'ençà que havíem descobert la possibilitat de reduir de 9 a 5 requadres 3×3 el rastreig de forats per tapar dins d'**ACTUALITZA-PLUS**: si creuen que la mateixa reducció es pot aplicar a **EXPLORA-3*9**, van ben errats, perquè després de tots els emplenaments esdevinguts a l'**ACTUALITZA** que precedeix aquesta funció (el principal i els automàtics), és molt probable que resultin afectats tots els requadres. L'estalvi que ha de compensar amb escreix el plus de dedicació que suposa multiplicar per tres el nombre d'unitats explorades (de fer-ho només en 9 requadres 3×3 a fer-ho en 9 files, 9 columnes i 9 requadres) sortirà de subsanar una badada imperdonable que havíem comès: la generació d'allò que podríem anomenar "plantilles de combinacions de 2, 3... 8 elements", i que pot interpretar-se en clau de **norma 2** o de **norma 3**, fins ara la realitzàvem amb cada execució d'**EXPLORA-3*9**; aquesta funció comptava el nombre de caselles lliures i **EXPL-SUBCONJ** s'encarregava tant de crear la llista de combinacions adient com de detectar si s'infringia la **norma 2** en algun dels subconjunts de llistes **(cadr 2R)** dels requadres 3×3 corresponents a aquestes combinacions (ambdues funcions estaven preparades per funcionar amb una o altra norma, però amb l'accés **(EXPL-SUBCONJ ())** la que efectvament s'aplicava era la **2**). Sovint resulta rendible revisar els temes perquè s'acaben descobrint coses que un havia passat per alt, i aquesta n'era una: no tenia cap sentit que, per a cadascun dels valors candidats a la casella actual i per a cadascun dels 9 requadres 3×3, anéssim recreant les mateixes plantilles, màxim quan la subsanació d'aquesta pífia ha posat en evidència que seu el cost en temps no era genys negligible; la decisió correcta és generar el conjunt de les 7 plantilles a l'inici d'**OMPLE-SUDOKU**, una vegada per sempre. Ho farem mitjançant la nova funció **INI-LLISTES**, i la llista **LLISTES** creada tindrà 7 subllistes-plantilla, el significat de les quals seria bo aclarir ara ja que, quan tocava (*Etapa prèvia: crear una solució, II*), no ho havíem fet amb un minim de detall.

Ja sabeu que, si es compleix la **norma 1**, quan una unitat tingui **M** caselles lliures també tindrà **M** llistes, en el conjunt de les quals apareixeran **M** valors diferents. La comprovació de la **norma 2** consistirà a considerar tots els subconjunts binaris, ternaris... **(M-1)**-aris de les **M** llistes, per veure si en algun subconjunt **N**-ari de llistes (**N < M**) apareixen menys de **N** valors diferents. A la plantilla corresponent al nombre **M** de llistes, aquests subconjunts **N**-aris es representen amb llistes de **N** valors **0**, **1**... **(M-1)**, amb el significat de **1**a, **2**a... **M**-èsima llista de la unitat. La comprovació de la **norma 3** hauria consistit a considerar totes les combinacions binàries, ternàries... **(M-1)**-àries formades amb els **M** valors, per veure si els valors d'alguna combinació **N**-ària (**N < M**) només apareixien en menys de **N** llistes. A la plantilla corresponent al nombre **M** de llistes, aquestes combinacions **N**-àries s'haurien representat amb llistes de **N** valors **0**, **1**... **(M-1)**, amb el significat de **1**r, **2**n... M-èsim valor present a la unitat (no pas dels valors **1**, **2**... **9** en si). Hem utilitzat intencionadament el condicional per referir-nos a la **norma 3** perquè, malgrat l'estalvi propiciat per la rectificació i per l'aprofitament del muntatge de cerca per files i columnes d'**ACTUALITZA-PLUS** per tal d'encabir-hi el dispositiu detector d'infraccions a totes les normes, alleugerirem aquest dispositiu, tant la nova presència en **ACTUALITZA-PLUS** com a **EXPLORA-3*9** (que seguirà circumscrita a l'exploració per requadres 3×3) limitant-lo a controlar la **norma 2** (malgrat això, en la present exposició seguirem parlant de **normes 2**/**3**): a **EXPL-SUBCONJ** evitarem bifurcacions innecessàries i en **ACTUALITZA-PLUS**, alliberats ja de l'obligació de desenvolupar el codi alternatiu adreçat a la **norma 3**, simplificarem la detecció d'infraccions de la **norma 1**, reduint-la a un apèndix del dispositiu tapaforats: la part subratllada de **(if (= K 1) (PLUS-N->P J) (if (= K 0) (setq FORA T) ...))**. Però abans de passar al detall d'alguns tripijocs que hem hagut de fer per adaptar **ACTUALITZA-PLUS** als nous objectius, no volem donar per acabat el tema de la llista **LLISTES** sense proporcionar al lector una referència més gràfica. Perquè se'n faci la idea, més enllà de la probablement hermètica descripció que en fèiem més amunt, direm que, per exemple, la plantilla corresponent a unitats amb 5 caselles lliures (5 llistes en el fragment **(cadr 2R)** que li correspongués) tindria aquest aspecte:

```
(((0 1) (0 2) (0 3) (0 4) (0 5) (1 2) (1 3) (1 4) (1 5) (2 3) (2 4) (2 5) (3 4)
  (3 5) (4 5))
 ((0 1 2) (0 1 3) (0 1 4) (0 1 5) (0 2 3) (0 2 4) (0 2 5) (0 3 4) (0 3 5)
  (0 4 5) (1 2 3) (1 2 4) (1 2 5) (1 3 4) (1 3 5) (1 4 5) (2 3 4) (2 3 5)
  (2 4 5) (3 4 5))
 ((0 1 2 3) (0 1 2 4) (0 1 2 5) (0 1 3 4) (0 1 3 5) (0 1 4 5) (0 2 3 4)
  (0 2 3 5) (0 2 4 5) (0 3 4 5) (1 2 3 4) (1 2 3 5) (1 2 4 5) (1 3 4 5)
  (2 3 4 5))
 ((0 1 2 3 4) (0 1 2 3 5) (0 1 2 4 5) (0 1 3 4 5) (0 2 3 4 5) (1 2 3 4 5)))
```

Tornant al nou **ACTUALITZA-PLUS**, a diferència del dispositiu de control de la **norma 1**, en què la condició **(if (= K 0) (setq FORA T) ...)** s'ha de complir amb tots els valors absents de la fila o la columna, la detecció d'infraccions a les **normes 2**/**3** s'ha de produir només una vegada, raó per la qual la seva integració en els bucles de cerca de forats haurà de fer-se de forma que només es posi en marxa en l'última iteració, just després de veure que el valor corresponent apareix més d'una vegada (en el lloc ocupat pels punts suspensius de l'expressió vista en l'últim paràgraf, **(if (= K 1) (PLUS-N->P J) (if (= K 0) (setq FORA T) ...))**). Per evitar que fila o columna puguin tenir un element **(nil nil nil nil nil nil nil nil nil)**, equivalent a **(car NIL*NIL)**, aprofitarem el primer terc d'**ACTUALITZA-PLUS** (el de rastreig de les caselles lliures amb només un valor admissible) per apartar del procés aquesta possibilitat, mitjançant l'activació de **FORA**. Que la detecció d'aquests elements **(car NIL*NIL)** de **(cadr 2R)** vagi a càrrec d'**ACTUALITZA-PLUS** < **ACTUALITZA**, permetrà eliminar de **RE-MEMBER** algunes giragonses que llastaven l'expressió

```
(progn
  (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
  (if FORA
      (setq FORA ())
      (if (equal (cadr 2R) NIL*NIL)
          (setq OK T)
          (progn
            (foreach F (cadr 2R)
              (if (and (not OK) (member (car NIL*NIL) F))
                  (setq OK T)))
            (if (not OK) (setq OK (EXPLORA-3*9)))
            (if OK
                (setq OK ())
                (RE-MEMBER (car 2R) (cadr 2R)))))))
```

que quedarà reduïda a

```
(progn
  (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
  (if (not FORA)
      (if (equal (cadr 2R) NIL*NIL)
          (setq OK T)
          (if (not (EXPLORA-3*9))
              (RE-MEMBER (car 2R) (cadr 2R))))))
```

Pel que fa a **ACTUALITZA**, retornarem la condició **(not (or PLUS FORA))** a la versió precedent **(not PLUS)**. A l'anterior **ACTUALITZA-PLUS** (oberta a la possibilitat que dos o més valors potencials apareguessin només un cop a la fila o columna i que corresponguessin a la mateixa casella, però sense anar més enllà), era possible que **PLUS** s'hagués activat provisionalment i hi hagués hagut assignació automàtica, però volíem que una posterior activació de **FORA** invalidés aquestes accions, raó per la qual havíem d'evitar nous accessos a **ACTUALITZA-PLUS**, entre altres coses. Era per això (perquè podia ser que **PLUS** = **T** i **FORA** = **T**, i que, després d'executar-se aquesta funció i d'invertir-se **PLUS**, hi hagués una segona volta) que estrenyíem l'accés passant de **(while (not PLUS) ...)** a **(while (not (or PLUS FORA)) ...)**, però ara això canviarà radicalment: ja no podrà ser **PLUS** = **T** i **FORA** = **T** (perquè amb la nova funció **ACTUALITZA-PLUS** només serà **FORA** = **T** si roman **PLUS** = **nil**), i no sols no caldrà afegir **(not FORA)** (pàgines abans ja havíem indicat que **(not (or PLUS FORA))** equival a **(and (not PLUS) (not FORA))**) sinó que per mantenir-ho hagués calgut fer també **(setq FORA ())** a l'inici d'**ACTUALITZA-PLUS**: altrament, a la segona part de **RE-MEMBER** en què **INI** = **nil**, després d'un **(ACTUALITZA R-9*9 E R-P R-LLL ())** o d'un **(EXPLORA-3*9)** que haguessin provocat **FORA** = **T** (per haver trobat una casella sense cap valor potencial o per infringir una fila, columna o requadre 3×3 la **norma 1**), a la previsible prohibició d'explorar més caselles lliures se n'hi hauria sumat la imprevista de no tenir més accés a **ACTUALITZA-PLUS** < **ACTUALITZA** amb altres valors candidats (de fet, aquest accident ja hagués pogut passar a la versió precedent).

Anem a presentar-vos el codi de les noves funcions i de les afectades per canvis, després de les quals trobareu l'anunciada relació de casos d'incompliment de les **normes 2/3**, sense solució de continuïtat. Però abans és obligat donar-vos una bona notícia, sense la qual no tindria massa sentit oferir-vos el nou codi: la millora del procediment de detecció d'infraccions a les normes esmentades (singularment la creació de la colecció de 7 plantilles **LLISTES**) ha pogut compensar l'exigència de multiplicar per tres el nombre de controls, perquè, en l'emplenament inicial, el temps que triga a omplir-se el caseller central 3×3 ha baixat de 18 a 15 segons.

```
(defun EXPL-SUBCONJ (/ LA)
  (setq K 2)
  (foreach N (nth (- (length L) 4) LLISTES)
    (if (not FORA)
        (progn
          (setq K (1+ K))
          (foreach F N
            (if (not FORA)
                (progn
                  (setq LA ())
                  (foreach E F
                    (foreach M (nth E L)
                      (if (not (member M LA)) (setq LA (cons M LA)))))
                  (if (< (length LA) K) (setq FORA T)))))))))

; Exploració per 5 requadres 3*3, 9 files i 9 columnes, amb detecció d'infraccions
; a les normes 2/3 en aquestes 18 darreres unitats.
(defun ACTUALITZA-PLUS (/ 3*3 LL3 I L M N NN MN X1 X2 Y1 Y2)
  (setq FORA ())
  (mapcar '(lambda (F-9*9 F-LLL)
             (if (not (or PLUS FORA))
                 (mapcar '(lambda (E-9*9 E-LLL)
                            (if (and (not (or PLUS FORA)) E-LLL)
                                (progn
                                  (setq K 0)
                                  (foreach E E-LLL
                                    (if (and (< K 2) E) (setq N E K (1+ K))))
                                  (if (= K 1)
                                      (PLUS-N->P E-9*9)
                                      (if (= K 0) (setq FORA T))))))
                         F-9*9 F-LLL)))
          A-9*9 A-LLL)
  (if (not (or PLUS FORA))
      (progn
        (setq X2 (* (/ X 3) 3) Y2 (* (/ Y 3) 3))
        (foreach K '(6 3 0) (if (/= K Y2) (setq L (cons (list X2 K) L))))
        (foreach K '(6 3 0) (setq L (cons (list K Y2) L)))
        (foreach P1 L
          (if (not PLUS)
              (progn
                (setq X1 (car P1) Y1 (cadr P1)
                      3*3 (SUB-X*Y A-9*9 P1 (list (+ X1 2) (+ Y1 2)))
                      LL3 (SUB-X*Y A-LLL P1 (list (+ X1 2) (+ Y1 2))))
                (foreach N L1A9
                  (if (not PLUS)
                      (progn
                        (setq K 0)
                        (mapcar '(lambda (F-3*3 F-LL3)
                                   (mapcar '(lambda (E-3*3 E-LL3)
                                              (if (member N E-LL3)
                                                  (setq K (1+ K) J E-3*3)))
                                           F-3*3 F-LL3))
                                3*3 LL3)
                        (if (= K 1) (PLUS-N->P J)))))))))
  (mapcar '(lambda (9*9 LL3)
             (if (not (or PLUS FORA)) ; (and (not PLUS) (not FORA))
                 (mapcar '(lambda (F-9*9 F-LL3)
                            (if (not (or PLUS FORA))
                                (progn
                                  (setq NN ())
                                  (foreach N L1A9
                                    (if (not (member N F-9*9))
                                        (setq NN (cons N NN))))
                                  (if (> (length NN) 0) ; També (> (length NN) 1)
                                      (progn
                                        (setq NN (reverse NN) MN (last NN))
                                        (foreach N NN
                                          (if (not PLUS)
```

```
                                                    (progn
                                                      (setq K 0)
                                                      (if (and INI (= N MN))
                                                          (setq L ()))
                                                      (mapcar '(lambda (E-9*9 E-LL3)
                                                                 (if (member N E-LL3)
                                                                     (setq K (1+ K)
                                                                           J E-9*9))
                                                                 (if (and INI E-LL3
                                                                          (= N MN))
                                                                     (progn
                                                                       (setq I ())
                                                                       (foreach M E-LL3
                                                                         (if M (setq I
                                                                                 (cons M
                                                                                   I))))
                                                                       (setq L (cons I L
                                                                                 )))))
                                                              F-9*9 F-LL3)
                                                      (if (= K 1)
                                                          (PLUS-N->P J)
                                                          (if (= K 0)
                                                              (setq FORA T)
                                                              (if (and INI (= N MN)
                                                                       (> (length L) 2))
                                                                  (EXPL-SUBCONJ)))))))))
                                                                                    )))
                              9*9 LL3)))
          (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L PLUS)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
              A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM" "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P)
                   A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (ACT-LLL X Y A-LLL))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

 (defun EXPLORA-3*9 (/ FORA R-9 R-L L LL I M)
  (foreach Y '(0 3 6)
    (foreach X '(0 3 6)
      (if (not FORA)
          (progn
            (setq R-9 (SUB-X*Y (car 2R) (list X Y) (list (+ X 2) (+ Y 2)))
                  R-L (SUB-X*Y (cadr 2R) (list X Y) (list (+ X 2) (+ Y 2))) L ())
            (foreach F R-9
              (foreach E F (if (atom E) (setq L (cons E L)))))
            (if (< (length L) 9) ; També ha de ser (< (length L) 8)
                (progn
                  (foreach F R-L
                    (foreach E F
                      (if E (foreach M E
                              (if (and M (not (member M L)))
                                  (setq L (cons M L)))))))
                  (if (< (length L) 9) (setq FORA T))
```

```
                    (if (and INI (not FORA))
                        (progn
                          (setq L ())
                          (foreach F R-L
                            (foreach E F
                              (if E (progn
                                      (setq I ())
                                      (foreach M E (if M (setq I (cons M I))))
                                      (setq L (cons I L))))))
                          (if (> (length L) 2) (EXPL-SUBCONJ))
                          (if (not FORA) (TERCETS R-9 R-L))))))
              (if (and INI (not FORA)) (setq N/R-L (3*3/R-L N/R-L)))))))
  FORA)

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ())
              INI (or FORA (EXPLORA-3*9)))
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
  (if (not (or INI OK))
      (if (not (setq OK (COMPLET-9*9 R-9*9)))
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (if (not FORA)
                        (if (equal (cadr 2R) NIL*NIL)
                            (setq OK T)
                            (if (not (EXPLORA-3*9))
                                (RE-MEMBER (car 2R) (cadr 2R))))))))))))
  OK)

(defun INI-LLISTES (/ LA LL)
  (foreach N '(3 4 5 6 7 8 9)
    (setq LA ())
    (repeat N (setq N (1- N) LA (cons (list N) LA)))
    (setq LL LA)
    (repeat (1- (length LL))
      (setq LA LL LL ())
      (repeat (1- (length LA))
        (setq I (car LA) LA (cdr LA) J (reverse (cdr (reverse I))))
        (foreach E LA
          (if (equal (reverse (cdr (reverse E))) J)
              (setq LL (cons (append I (list (last E))) LL)))))
      (setq LL (reverse LL) LLL (cons LL LLL)))
    (setq LLL (reverse (cdr LLL)))
    (repeat (- (length (last LLL)) 3) (setq LLL (cdr LLL)))
    (setq LLISTES (cons LLL LLISTES)))
  (setq LLISTES (reverse LLISTES)))

(defun OMPLE SUDOKU (/ LLISTES 0*0 NIL*NIL CURSOR P-CURS C->F INV-F JJ JJ1 JJ2 TT
                       N/R-L TN/R-L)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (INI-LLISTES)
        (INI-LLL ())
        (setq NIL*NIL LLL ...) ...) ...) ...)
```

0 caselles ocupades (9 llistes):

- 2 valors en només 1 llista - 8 llistes amb només 7 valors

- 3 valors en només 2 llistes- 7 llistes amb només 6 valors
- 3 valors en només 1 llista - 8 llistes amb només 6 valors

- 4 valors en només 3 llistes- 6 llistes amb només 5 valors
- 4 valors en només 2 llistes- 7 llistes amb només 5 valors
- 4 valors en només 1 llista - 8 llistes amb només 5 valors

- 5 valors en només 4 llistes- 5 llistes amb només 4 valors
- 5 valors en només 3 llistes- 6 llistes amb només 4 valors
- 5 valors en només 2 llistes- 7 llistes amb només 4 valors
- 5 valors en només 1 llista - 8 llistes amb només 4 valors

- 6 valors en només 5 llistes- 4 llistes amb només 3 valors
- 6 valors en només 4 llistes- 5 llistes amb només 3 valors
- 6 valors en només 3 llistes- 6 llistes amb només 3 valors
- 6 valors en només 2 llistes- 7 llistes amb només 3 valors
- 6 valors en només 1 llista - 8 llistes amb només 3 valors

- 7 valors en només 6 llistes- 3 llistes amb només 2 valors
- 7 valors en només 5 llistes- 4 llistes amb només 2 valors
- 7 valors en només 4 llistes- 5 llistes amb només 2 valors
- 7 valors en només 3 llistes- 6 llistes amb només 2 valors
- 7 valors en només 2 llistes- 7 llistes amb només 2 valors
- 7 valors en només 1 llista - 8 llistes amb només 2 valors

- 8 valors en només 7 llistes- 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)
- 8 valors en només 6 llistes- 3 llistes amb només 1 valor (no)
- 8 valors en només 5 llistes- 4 llistes amb només 1 valor (no)
- 8 valors en només 4 llistes- 5 llistes amb només 1 valor (no)
- 8 valors en només 3 llistes- 6 llistes amb només 1 valor (no)
- 8 valors en només 2 llistes- 7 llistes amb només 1 valor (no)
- 8 valors en només 1 llista - 8 llistes amb només 1 valor (no)

1 casella ocupada (8 llistes):

- 2 valors en només 1 llista - 7 llistes amb només 6 valors

- 3 valors en només 2 llistes- 6 llistes amb només 5 valors
- 3 valors en només 1 llista - 7 llistes amb només 5 valors

- 4 valors en només 3 llistes- 5 llistes amb només 4 valors
- 4 valors en només 2 llistes- 6 llistes amb només 4 valors
- 4 valors en només 1 llista - 7 llistes amb només 4 valors

- 5 valors en només 4 llistes- 4 llistes amb només 3 valors
- 5 valors en només 3 llistes- 5 llistes amb només 3 valors
- 5 valors en només 2 llistes- 6 llistes amb només 3 valors
- 5 valors en només 1 llista - 7 llistes amb només 3 valors

- 6 valors en només 5 llistes- 3 llistes amb només 2 valors
- 6 valors en només 4 llistes- 4 llistes amb només 2 valors
- 6 valors en només 3 llistes- 5 llistes amb només 2 valors
- 6 valors en només 2 llistes- 6 llistes amb només 2 valors
- 6 valors en només 1 llista - 7 llistes amb només 2 valors

- 7 valors en només 6 llistes- 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)
- 7 valors en només 5 llistes- 3 llistes amb només 1 valor (no)
- 7 valors en només 4 llistes- 4 llistes amb només 1 valor (no)
- 7 valors en només 3 llistes- 5 llistes amb només 1 valor (no)
- 7 valors en només 2 llistes- 6 llistes amb només 1 valor (no)
- 7 valors en només 1 llista - 7 llistes amb només 1 valor (no)

2 caselles ocupades (7 llistes):

- 2 valors en només 1 llista - 6 llistes amb només 5 valors

- 3 valors en només 2 llistes- 5 llistes amb només 4 valors
- 3 valors en només 1 llista - 6 llistes amb només 4 valors

- 4 valors en només 3 llistes- 4 llistes amb només 3 valors
- 4 valors en només 2 llistes- 5 llistes amb només 3 valors
- 4 valors en només 1 llista - 6 llistes amb només 3 valors

- 5 valors en només 4 llistes- 3 llistes amb només 2 valors
- 5 valors en només 3 llistes- 4 llistes amb només 2 valors
- 5 valors en només 2 llistes- 5 llistes amb només 2 valors
- 5 valors en només 1 llista - 6 llistes amb només 2 valors

- 6 valors en només 5 llistes- 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)
- 6 valors en només 4 llistes- 3 llistes amb només 1 valor (no)
- 6 valors en només 3 llistes- 4 llistes amb només 1 valor (no)
- 6 valors en només 2 llistes- 5 llistes amb només 1 valor (no)
- 6 valors en només 1 llista - 6 llistes amb només 1 valor (no)

3 caselles ocupades (6 llistes):

- 2 valors en només 1 llista - 5 llistes amb només 4 valors

- 3 valors en només 2 llistes- 4 llistes amb només 3 valors
- 3 valors en només 1 llista - 5 llistes amb només 3 valors

- 4 valors en només 3 llistes- 3 llistes amb només 2 valors
- 4 valors en només 2 llistes- 4 llistes amb només 2 valors
- 4 valors en només 1 llista - 5 llistes amb només 2 valors

- 5 valors en només 4 llistes- 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)
- 5 valors en només 3 llistes- 3 llistes amb només 1 valor (no)
- 5 valors en només 2 llistes- 4 llistes amb només 1 valor (no)
- 5 valors en només 1 llista - 5 llistes amb només 1 valor (no)

4 caselles ocupades (5 llistes):

- 2 valors en només 1 llista - 4 llistes amb només 3 valors

- 3 valors en només 2 llistes- 3 llistes amb només 2 valors
- 3 valors en només 1 llista - 4 llistes amb només 2 valors

- 4 valors en només 3 llistes- 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)
- 4 valors en només 2 llistes- 3 llistes amb només 1 valor (no)
- 4 valors en només 1 llista - 4 llistes amb només 1 valor (no)

5 caselles ocupades (4 llistes):

- 2 valors en només 1 llista - 3 llistes amb només 2 valors

- 3 valors en només 2 llistes- 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)
- 3 valors en només 1 llista - 3 llistes amb només 1 valor (no)

6 caselles ocupades (3 llistes):

- 2 valors en només 1 llista - 2 llistes amb només 1 valor (no: exclòs de **LLISTES**)

Prescindim del grup 7 caselles ocupades (2 llistes) perquè l'únic cas conflictiu, un valor que no ocupés cap casella de la unitat (fila, columna o requadre 3×3) ni tampoc fos admissible en primera instància a cap casella buida (hauria figurat com

- 1 valor en 0 llistes- 2 llistes amb només 1 valor), seria detectat per infracció de la **norma 1** (en **ACTUALITZA-PLUS** o **EXPLORA-3*9**, en aquesta última funció només en el supòsit que l'incompliment no es produís a nivell de files ni de columnes, però sí en algun requadre 3×3). Tot i que, a la pràctica, seria **ACTUALITZA-PLUS** qui ho detectaria primer, mitjançant un procés bastant rocambolesc: en una primera etapa emplenaria amb el valor admissible únic la casella que localitzés abans i, en una segona etapa (recordeu que l'activació de **PLUS** desencadena una nova intervenció de la funció) es trobaria l'altra amb un **E-LLL (nil nil nil nil nil nil nil nil nil)**. Però ben mirat també podríem prescindir de l'últim grup que figurava a la relació, 6 caselles ocupades (3 llistes), perquè el cas conflictiu (- 2 valors en només 1 llista - 2 llistes amb només 1 valor) tampoc no arribarà a accedir a **EXPL-SUBCONJ**, pel mateix motiu: la intervenció d'**ACTUALITZA-PLUS** en la primera casella buida hi col·locarà l'únic valor admissible i activarà **PLUS**; la segona intervenció forçada per **PLUS** es trobarà que la segona casella ja no admet cap valor, activarà **FORA** i el valor candidat a la casella actual serà refusat. Per la mateixa raó, dels grups precedents tampoc no accediran a **EXPL-SUBCONJ** els casos de 2, 3... 8 llistes amb només 1 valor, i per això els hem posat l'epíleg "(no)". D'aquesta possibilitat i de la manera com els dispositius previs se'n desempallegarien ja n'havíem parlat en referir-nos a unitats (files, columnes o requadres 3×3) amb 8 valors en només 7 llistes (- 2 llistes amb només 1 valor), quan pàgines enrera estàvem centrats en el grup 0 caselles ocupades (9 llistes) sense ser-ne del tot conscients. Una altra història és que a la plantilla **LLISTES** només podíem eliminar el primer membre i la primera sublista dels demés (combinacions binàries de llistes **E-LLL**), per reduir la cerca. A la versió d'**INI-LLISTES** que recull aquesta simplifició, hem subratllat els canvis en relació a la precedent (la de l'últim bloc de codi presentat):

```
(defun INI-LLISTES (/ LA LL)
  (foreach N '(3 4 5 6 7 8 9)
    (setq LA ())
    (repeat N (setq N (1- N) LA (cons (list N) LA)))
    (setq LL LA)
    (repeat (1- (length LL))
      (setq LA LL LL ())
      (repeat (1- (length LA))
        (setq I (car LA) LA (cdr LA) J (reverse (cdr (reverse I))))
        (foreach E LA
          (if (equal (reverse (cdr (reverse E))) J)
              (setq LL (cons (append I (list (last E))) LL)))))
      (setq LL (reverse LL) LLL (cons LL LLL)))
    (setq LLL (reverse (cdr LLL)))
    (repeat (- (length (last LLL)) 3) (setq LLL (cdr LLL)))
    (setq LLISTES (cons (cdr LLL) LLISTES)))
  (setq LLISTES (cdr (reverse LLISTES))))
```

A conseqüència d'aquesta simplificació, també caldrà substituir uns valors de les funcions següents:

- A **EXPL-SUBCONJ**, cal substituir
  (setq K 1)
  (foreach N (nth (- (length L) 3) LLISTES)
  (2ª i 3ª línia de codi) per
  (setq K 2)
  (foreach N (nth (- (length L) 4) LLISTES)

- En **ACTUALITZA-PLUS**, cal substituir
  (> (length L) 2))
  (76ª línia de codi) per
  (> (length L) 3))

- A **EXPLORA-3*9**, cal substituir
  (if (> (length L) 2) (EXPL-SUBCONJ))
  (27ª línia de codi) per
  (if (> (length L) 3) (EXPL-SUBCONJ))

Ara que ho tenim gairebé tot controlat, seria l'hora de recapitular i veure si els dispositius que intervenen en el sentit d'aportar "intel·ligència" a la cerca cega de **RE-MEMBER**, anticipant el veredicte sobre la viabilitat dels candidats a ocupar la casella actual (el dispositiu tapaforats, que contribueix a avançar veredictes

tant positius com negatius, i la detecció d'infraccions a les **normes 1** -sempre- i **2**/**3** -només quan **INI** = **T**- que, conjuntament amb la detecció de caselles buides amb expectatives nul·les, sols contribueix a avançar els negatius) estan organitzats o no de forma eficient, tenint en compte que s'han anat implementant sobre la marxa, en una progressió que podem qualificar de moltes maneres menys de poc accidentada: en són prou evidència els quatre subcapítols (de moment) en què hem subdividit el capítol *Etapa prèvia: cear una solució* que, a còpia d'ensopegades, no ha parat de créixer. Aquest és el resum de competències d'**ACTUALITZA-PLUS** i **EXPLORA-3*9**, tal i com els tenim ara, presentades per ordre d'intervenció:

- **ACTUALITZA-PLUS**, pot ser amb **(not REAL)** (des de **RE-MEMBER**, amb **INI** o **(not INI)**) o amb **REAL** (i **(not INI)**, quan no s'han apreciat configuracions **AAA**-**BBB** i ja se sap que el valor candidat és viable, es repeteixen les execucions de **RE-MEMBER** **INI**, però només la part tapaforats):
  - En l'àmbit de tot l'escaquer 9×9:
    - Si una casella té un únic valor potencial, assignació d'aquest valor.
    - Si una casella no té cap valor potencial, expulsió del nivell d'autorecursió actual de **RE-MEMBER**.
  - Per requadres 3×3 (només amb els 5 afectables per l'emplenament en la casella actual), assignació automàtica de valors potencials únics en tot el requadre.
  - Per files (9):
    - Assignació automàtica de valors potencials únics en tota la fila.
    - Si la fila infringeix la **norma 1**, expulsió del nivell d'autorecursió actual de **RE-MEMBER**.
    - Si **INI** = **T** i la fila infringeix les **normes 2**/**3**, expulsió de **RE-MEMBER**.
  - Per columnes (9):
    - Assignació automàtica de valors potencials únics en tota la columna.
    - Si la columna infringeix **norma 1**, expulsió del nivell d'autorecursió actual de **RE-MEMBER**.
    - Si **INI** = **T** i la columna infringeix les **normes 2**/**3**, expulsió de **RE-MEMBER**.

- **EXPLORA-3*9** (només des de **RE-MEMBER**, amb **INI** o **(not INI)**):
  - Per requadres 3×3 (9):
    - Si el requadre infringeix **norma 1**, expulsió del nivell d'autorecursió actual de **RE-MEMBER**.
    - Si **INI** = **T**:
      - Si el requadre infringeix les **normes 2**/**3**, expulsió de **RE-MEMBER**.
      - Si no, cerca de les condicions necessàries perquè hi hagi configuracions **AAA**-**BBB** (si es donen, ja des de **RE-MEMBER**, s'explorarà per requadres 9×3 i 3×9 a la cerca de les condicions suficients i, si s'escau, se simplificarà la pseudomatriu **A-LLL** de valors potencials).

Tenint en compte la gènesi certament atzarosa d'aquesta versió, encara ens podem vanagloriar de tenir un producte prou estructurat. Potser quedaria més estètic que haguéssim optat per aplegar tots els dispositius tapaforats en una sola funció que actués per requadres 3×3, files i columnes, i l'exploració a la caça d'infraccions a la **norma 1** i a les **normes 2**/**3** (només a l'etapa **INI**cial de **RE-MEMBER**) també per requadres 3×3, files i columnes, però no ens podem queixar de la racionalitat del que tenim, si més no des del criteri d'optimització d'ús dels bucles d'exploració. És evident que, si haguéssim desglossat les exploracions per funcions, fent d'una banda **ACTUALITZA-PLUS** (**A**), amb
- exploració de 9 files, sense mantenir informació de casella a casella (**A**)
- exploració de 5 requadres 3×3, mantenint-hi informació de les 9 caselles (**A**)
- exploració de 9 files, mantenint-hi informació de les 9 caselles (**A**)
- exploració de 9 columnes, mantenint-hi informació de les 9 caselles (**A**)

i de l'altra **EXPLORA-3*9** (**E**), amb
- exploració de 9 requadres 3×3, mantenint-hi informació de les 9 caselles (**E**)
- exploració de 9 files, mantenint-hi informació de les 9 caselles (**E**)
- exploració de 9 columnes, mantenint-hi informació de les 9 caselles (**E**)

no sortiríem guanyant en relació a l'hibrid **ACTUALITZA-PLUS** + **EXPLORA-3*9** actual,
- exploració de 9 files, sense mantenir informació de casella a casella (**A**)
- exploració de 5 requadres 3×3, mantenint-hi informació de les 9 caselles (**A**)
- exploració de 9 files, mantenint-hi informació de les 9 caselles (**A**+**E**)
- exploració de 9 columnes, mantenint-hi informació de les 9 caselles (**A**+**E**)
- exploració de 9 requadres 3×3, mantenint-hi informació de les 9 caselles (**E**)

perquè organitzar la informació per requadres 3×3 (funció **SUB-X*Y**) i passar de files a columnes (funció **TRANSPOSAR**) té un cost en temps, i és pitjor muntar les estructures de bucles específiques set vegades que cinc.

Però, prioritzant aquest criteri, no negareu que encara és millor ordenar-ho així:
- exploració de 9 files, mantenint-hi informació de les 9 caselles (**A**+**A**+**E**)
- exploració de 9 columnes, mantenint-hi informació de les 9 caselles (**A**+**E**)
- exploració de 9 requadres 3×3 (4 exclosos de l'acció **A**), mantenint-hi informació de les 9 caselles (**A**+**E**).

Tot i que al final de la penúltima pàgina hem deixat caure un inquietant gairebé subratllat, com donant a entendre que encara no teníem tots els caps ben lligats (i encara més enrere, en la pàgina 300, s'insinuava la necessitat de dedicar un cinquè subcapítol a resoldre un problema pendent), no seria recomanable deixar per al final una repassada general per depurar el codi: conscients que podem compactar dispositius d'origen dispar però capaços de compartir un mateix sostre, fem-ho ara i, quan haguem de fer front a nous conflictes, tindrem el terreny més desbrossat.

Si, dintre del nou esquema mantenim el mateix ordre d'operacions i atenem sobretot a criteris d'eficàcia en el processat d'informació, és perquè fins a cert punt la posició relativa de la major part de dispositius és indiferent. Per exemple, les dues etapes de cerca de forats per tapar, caselles amb només un valor admissible i caselles que tenen un valor predeterminat per ser les úniques que l'admeten en una unitat modular (requadre 3×3, fila o columna), si convingués es podrien permutar i no passaria res de res. Aquesta afirmació és evident quan els forats de les dues categories no comparteixen cap unitat modular, però no ho resulta tant en altres circumstàncies, raó per la qual serà bo fer una petita marrada per comprovar com tots els camins porten a Roma. Per posar-ho difícil, imaginem una columna amb sols dues caselles **P** i **Q** que admeten un valor **n**, però mentre que **P** no n'admet cap més **Q** admet altres valors, tot i ser l'única de la seva fila compatible amb **n** (si també ho és en el seu requadre 3×3, o el fet que aquest contingui o no la casella **P**, són qüestions que no afectaran la nostra argumentació): si comencem amb la cerca dels forats de categoria **Q**, **n** s'assignarà a aquesta casella, la columna comú a **Q** i **P** s'actualitzarà amb la desaparició del valor **n** de la llista **(cadr 2R)** corresponent a la segona, que quedarà reduïda a **(nil nil nil nil nil nil nil nil nil)**, activarà **FORA** i interromprà l'exploració de la candidatura en curs, que serà desestimada; si comencem amb la cerca dels forats de categoria **P**, com hem fet i seguirem fent per conveniències de processat, **n** s'assignarà a aquesta casella, la columna comú a **P** i **Q** s'actualitzarà amb la desaparició del valor **n** de la llista corresponent a la segona (única de la seva fila on teníem aquest valor), fet que infringirà la **norma 1**, activarà **FORA** i interrompirà l'exploració de la candidatura en curs, que també acabarà desestimada. Observeu que l'ambivalència en l'ordre d'exploració s'estén a totes les recursions, perquè les úniques accions limitades a la recursió **INI**cial són el control de les **normes 2**/**3** i la detecció de configuracions **AAA-BBB**.

Tammateix, davant d'alguns canvis convindrà explicar per què s'han realitzat tal i com ens mostra el codi i no d'una altra manera. Per exemple, posats a aplegar les exploracions per requadres 3×3, a algú se li podia ocórrer que era més senzill dur **EXPLORA-3*9** al terç central d'**ACTUALITZA-PLUS**, aprofitant l'estructura de cerca de només els 5 requadres afectats per la casella activada o per cadascun dels forats emplenats automàticament a conseqüència d'aquesta, en comptes de desplaçar cap al final les exploracions tapaforats per requadres 3×3 i estendre el control de les **normes 2**/**3** a tots 9 requadres (cosa que ja havíem justificat per la necessitat de fer front al cas més desfavorable: que s'hagués produït una cadena d'emplenaments automàtics que afectés a totes les regions de l'escaquer 9×9). L'objecció a un tal suggeriment no es podria despatxar expeditivament dient que és millor limitar-se a explorar 5 requadres que fer-ho en 9, òbviament, tret que això comportés repetir a cada forat tapat aquest **EXPLORA-3*9** d'abast més curt, perquè tampoc no està massa clar que calgui una comprovació de les **normes 1**, **2** i **3** després de cada forat tapat (no tenint cap garantia que no n'hi haurà cap més) o que n'hi hagi prou a fer-ho una vegada (prèvia a la seqüència d'emplenaments automàtics, entre emplenament i emplenament, o després de l'últim). Perquè d'una banda tot sembla apuntar a que l'estatus que tingui l'escaquer 9×9 de cara al compliment o no de les **normes 1**, **2** i **3**, just abans d'una sèrie ininterrompuda d'emplenaments automàtics, s'hauria de mantenir al final, i que qualsevol alteració d'aquest estatus hauria de venir d'un emplenament discrecional (virtual o real) i no pas d'un esdeveniment ineluctable (l'emplenament d'una casella amb un valor predeterminat). Però de l'altre n'estem fins al capdamunt de veure com un emplenament automàtic precipita una infracció de la **norma 1** (l'últim cop, en el paràgraf precedent), i el propòsit del subcapítol que ens veurem obligats a afegir consistirà precisament a fer aflorar determinades relacions que ja es donen amb l'ultim emplenament discrecional, de forma larvada, però que necessiten d'un tractament per mostrar les infraccions de les **normes 2**/**3**.

Però no té sentit seguir especulant sobre el tema perquè, per damunt d'aquestes i altres consideracions (també podríem dir que a efectes pràctics seria millor que primerament actués **EXPLORA-3*9** i després **ACTUALITZA-PLUS**, ja que d'aquesta manera s'evitaria perdre temps tapant forats a partir d'emplenaments discrecionals que un **EXPLORA-3*9** ampliat a files, columnes i requadres podria desqualificar d'entrada), la presència de **TERCETS** redueix el ventall d'opcions a una de sola: quan **INI** = **T**, **TERCETS** ha d'intervenir quan la cadena d'emplenaments automàtics hagi arribat al seu fi i estigui comprovat el compliment de les **normes 1**, **2** i **3**, i no només perquè convenia dur els emplenaments automàtics el més lluny possible perquè així era més senzill identificar les eventuals configuracions **AAA-AAA** i **AAA-BBB**, sinó perquè el muntatge basat en la cerca d'aquestes configuracions per requadres 3×3 (**TERCETS**), verificació per files o columnes (**FILS/COLS**) i, si s'escau, substitució a **R-LLL** (**REORD**), parteix d'aquesta premisa. Veieu com queden **ACTUALITZA-PLUS** i **RE-MEMBER**:

```
(defun PLUS-N->P (M / P1 P2)
  (setq A-N Ñ A-P M PLUS T)
  (if REAL
      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa Ñ))
      (if INI (setq REPLUS (ACT-9*9 Ñ M REPLUS)))))

(defun EXPLORA-9+9+5 (/ NN MN)
  (foreach Ñ L1A9 (if (not (member Ñ F-9*9)) (setq NN (cons Ñ NN))))
  (if (> (length NN) 0)
      (progn
        (setq NN (reverse NN) MN (last NN))
        (foreach Ñ NN
          (if (not PLUS)
              (progn
                (setq K 0)
                (if (and INI (= Ñ MN)) (setq L ()))
                (mapcar '(lambda (E-9*9 E-LL3)
                           (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
                           (if (and INI E-LL3 (= Ñ MN))
                               (progn
                                 (setq I ())
                                 (foreach M E-LL3 (if M (setq I (cons M I))))
                                 (setq L (cons I L)))))
                        F-9*9 F-LL3)
                (if (= K 1)
                    (PLUS-N->P J)
                    (if (= K 0)
                        (setq FORA T)
                        (if (and INI (= Ñ MN) (> (length L) 3))
                            (EXPL-SUBCONJ)))))))))

(defun ACTUALITZA-PLUS (/ I L M Ñ X1 Y1 5P R-9 R-L ORIG)
  (setq FORA ())
  (mapcar '(lambda (9*9 LL3)
             (setq ORIG (not ORIG))
             (if (not (or PLUS FORA))
                 (mapcar '(lambda (F-9*9 F-LL3)
                            (if (and (not (or PLUS FORA)) ORIG)
                                (mapcar '(lambda (E-9*9 E-LLL)
                                           (if (and (not (or PLUS FORA)) E-LLL)
                                               (progn
                                                 (setq K 0)
                                                 (foreach E E-LLL
                                                   (if (and (< K 2) E)
                                                       (setq Ñ E K (1+ K))))
                                                 (if (= K 1)
                                                     (PLUS-N->P E-9*9)
                                                     (if (= K 0)
                                                         (setq FORA T))))))
                                        F-9*9 F-LL3))
                            (if (not (or PLUS FORA)) (EXPLORA-9+9+5)))
                          9*9 LL3)))
           (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL)))
```

```
  (if (not (or PLUS FORA))
      (progn
        (setq X1 (* (/ X 3) 3) Y1 (* (/ Y 3) 3) ORIG ())
        (foreach K '(6 3 0) (if (/= K Y1) (setq 5P (cons (list X1 K) 5P))))
        (foreach K '(6 3 0) (setq 5P (cons (list K Y1) 5P)))
        (repeat (if INI 2 1)
          (setq ORIG (not ORIG))
          (if (not (or PLUS FORA))
              (foreach Y '(0 3 6)
                (foreach X '(0 3 6)
                  (if (and ORIG (not (or PLUS FORA)) (member (list X Y) 5P))
                      (progn
                        (setq R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              5P (subst (cons (list X Y) (list R-9 R-L))
                                                         (list X Y) 5P))
                        (mapcar '(lambda (F-9*9 F-LL3)
                                   (if (not (or PLUS FORA)) (EXPLORA-9+9+5)))
                                (list (append (car R-9) (cadr R-9) (last R-9)))
                                (list (append (car R-L) (cadr R-L) (last R-L))))))
                  (if (not ORIG)
                      (progn
                        (if (setq R-L (assoc (list X Y) 5P))
                            (setq R-9 (cadr R-L) R-L (caddr R-L))
                            (setq R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))
                                  R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))))
                        (setq K 0)
                        (foreach F R-9
                          (foreach E F (if (atom E) (setq K (1+ K)))))
                        (if (< K 9) (TERCETS R-9 R-L))
                        (setq N/R-L (3*3/R-L N/R-L)))))))))))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
  (if (not (or INI OK))
      (if (not (setq OK (COMPLET-9*9 R-9*9)))
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (if (not FORA)
                        (if (equal (cadr 2R) NIL*NIL)
                            (setq OK T)
                            (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)
```

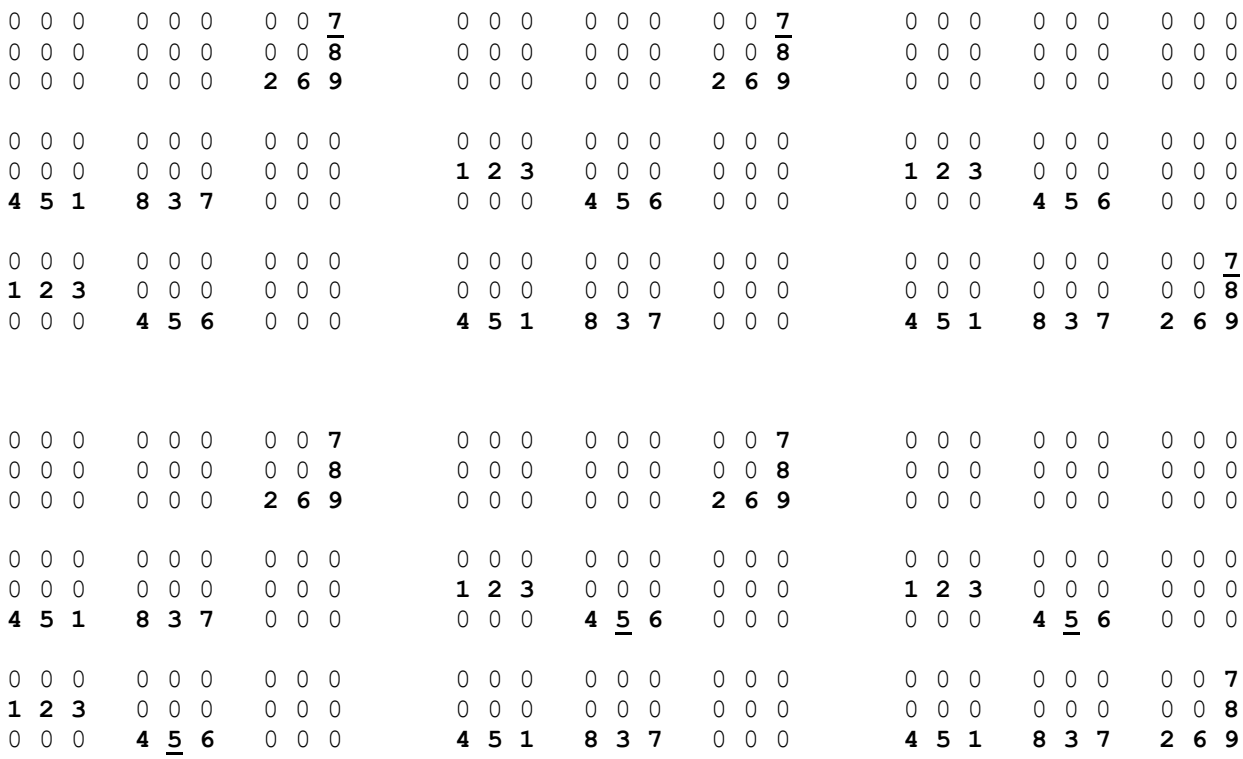
Com era previsible, en tractar d'aplegar l'acció tapaforats i de control de normes en els requadres 3×3 s'ha posat de manifest que la divisió entre **ACTUALITZA-PLUS** i **EXPLORA-3*9** era una decisió conjuntural que ara esdevé artificiosa i innecessària: la primera funció ha acabat fent-se càrrec de les dues, **EXPLORA-3*9** ha desaparegut i en el seu lloc hem creat la funció **EXPLORA-9+9+5**, subordinada a **ACTUALITZA-PLUS** i que, com suggereix el nom, s'encarrega de rastrejar les 9 files, les 9 columnes i els 5 requadres afectats per la casella actual i pels forats derivats d'aquesta.

És clar que el procés de compactació ha obligat a canvis subtils com, per exemple, a substituir en **ACTUALITZA-PLUS** la variable **N** per **Ñ** (circumstància que en afectar també a **PLUS-N->P** justifica que haguem inclòs aquesta funció entre les canviades) perquè, si no ho fèiem així, en els accessos a **TERCETS** i a **3*3/R-L** que té al final **ACTUALITZA-PLUS, N** (que en aquestes funcions representa el candidat examinat per **RE-MEMBER**) hauria estat confós amb l'últim valor testejat a **EXPLORA-9+9+5**. Però ha permès que en sortíssim amb un balanç positiu, perquè quan no ha estat possible de reduir el nombre de bucles (com a la primera meitat d'**ACTUALITZA-PLUS**, en què la cerca de caselles amb només un valor admissible es fa aprofitant l'estructura de lectura per files que serveix per localitzar les que tenen un valor predeterminat en ser les úniques de la fila que admeten un valor) hem conservat informació del bucle precedent (com a la segona meitat, en què a 5 dels 9 requadres 3×3 hem pogut prescindir de **SUB-X*Y** per accedir al contingut d'aquestes unitats modulars).

Fixeu-vos que la decisió de donar prioritat a la velocitat de processat, unificant les estructures de rastreig segons l'esquema de tres fulls enrera, que repetim,
- exploració de 9 files, mantenint-hi informació de les 9 caselles (**A**+**A**+**E**)
- exploració de 9 columnes, mantenint-hi informació de les 9 caselles (**A**+**E**)
- exploració de 9 requadres 3×3 (4 exclosos de l'acció **A**), mantenint-hi informació de les 9 caselles (**A**+**E**)

en alguns casos podria no ser l'òptima. Per exemple, en un emplenament automàtic a l'àmbit d'un requadre 3×3, aquesta ordenació d'operacions comportarà que l'accés a **EXPL-SUBCONJ** s'hagi de realitzar 5 vegades: abans de produir-se l'emplenament, per files i columnes; després, si no n'hi ha cap més, per files, columnes i requadres. Amb la penúltima disposició (**EXPLORA-3*9** diferenciada d'**ACTUALITZA** i executada a continuació), en canvi, només hi accedíem 3 cops: per files, columnes i requadres. Però en tot allò que concerneixi a rendiment, els resultats experimentals compten més que les consideracions *a priori*, i en comparar les dues versions els resultats són clars: si l'emplenament de les primeres caselles és el més exasperant pel que fa al temps que triguen a sortir els valors candidats en el caseller central 3×3, hem aconseguit reduir-lo a dos terços (en el primer, els 15 segons que trigava a aparèixer el conjunt dels 9 candidats ha baixat a 10 segons, i aquesta proporció es manté ben bé en els primers 20 emplenaments).

I, per no deixar coses pendents, acabarem desvetllant-vos que aquelles anomalies que a l'inici del subcapítol (pàg. 243) se'ns superposaven a les que llavors vam començar a analitzar específicament (problemàtica de les configuracions **AAA-AAA**, **AAA-AAA**, **AAA-BBB** i **AAA-BBB**) i que, sense saber exactament a què responien, havíem volgut esquivar, per tal d'abordar en millors condicions les segones (a diferència d'aquestes, associades a veredictes negatius que trigaven moltíssim a arribar, les primeres tenien a veure amb veredictes favorables), ara ja les tenim sota control.



Per no haver de retrocedir 72 pàgines, us hem repetit aquelles sis representacions i us recordarem com es plantejava la qüestió: tant la 1ª renglera (**AAA-BBB**) com la segona (**AAA-BBB**) tenien la particularitat que el veredicte sobre el candidat de la casella que anàvem a emplenar (negatiu) era quasi instantani a les de l'esquerra, però en les del centre (en què els dos primers requadres 9×3 estaven permutats) i dreta (en què la casella actual i les quatre acompanyants ocupades es desplaçaven 6 posicions cap avall) l'espera del veredicte s'eternitzava. Doncs si fallaven les dues, ¿quina diferència hi havia entre les representacions del mig i de la dreta?: a la primera renglera, si deixàvem per al final l'emplenament del requadre 3×3 que contenia la candidatura maleïda **7**, en la del mig alguns candidats de les caselles 9,7 (**7**, **8** i **9**) i 9,8 (**7** i **8**), tots finalment admessos, també trigaven a aparèixer, igual que els candidats que en 9,9 precedien **7** (**1**, **3**, **4** i **5**), tot i que la demora d'aquest **7** era incomparablement més llarga, i tanmateix en la representació dreta només grinyolava aquest veredicte; anàlogament, si a la segona renglera deixàvem per al final l'emplenament del requadre 3×3 que contenia la candidatura maleïda **5**, en la del mig els candidats que la precedien (**1** i **2**) també trigaven, tot i que la demora d'aquest **5** era incomparablement més llarga, i tanmateix en la representació dreta únicament grinyolava aquest veredicte. Doncs bé, el tipus d'incidències que causaven aquests efectes col·laterals en les representacions del centre han de ser forçosament de les tractades en les últimes pàgines, però no us avorrirem volent-les localitzar. N'hi haurà prou a dir que hem arribat a aquesta conclusió afegint a l'última versió **RE-MEMBER** la línia subratllada, cosa que desmunta tot l'operatiu que llavors havíem organitzat per explicitar tercets implícits en configuracions **AAA-BBB** i **AAA-BBB**: utilitzant aquest esguerro improvisat en els casos representats al mig, les maleïdes candidatures **7** i **5** tornaran a obligar-nos a esperar asseguts el veredicte d'incompatibilitat, però no les precediran altres esperes menors, com no passa a la dreta. Si ho voleu comprovar, aquí teniu aquesta **RE-MEMBER** *ad hoc*:

```
(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TN/R-L ()
                    TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
  (if (not (or INI OK))
      (if (not (setq OK (COMPLET-9*9 R-9*9)))
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (if (not FORA)
                        (if (equal (cadr 2R) NIL*NIL)
                            (setq OK T)
                            (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)
```

Però abans de cantar victòria i enganyar-vos pensant que tot ja és sota control, recordeu aquell inquietant gairebé de la pàgina 310. Desvelarem el misteri passant plana (expressió que en aquest cas, per desgràcia, no és equivalent a girar full).

Etapa prèvia: crear una solució, V (la Seca, la Meca i la vall d'Andorra)

Heu passat plana i seguiu amb aquesta crònica de mai acabar, que ja està resultant francament enfadosa. Per enèsima vegada, la celebració d'un èxit que suposàvem que seria definitiu ha encallat, i en dues ensopegades s'ha passat de marxa triomfal a marxa fúnebre. Si heu estat atents a la selecció d'exemples d'infracció per files de les **normes 2**/**3**, la segona us la podíeu ensumar (mirant de trobar-ne alguna amb 7 valors en només 6 llistes de **(cadr 2R)**, el conflicte havia saltat prematurament i no ens vam estar de dir que l'explicació vindria en el subcapítol on som ara), però la primera s'ha produït on ningú no s'ho pensava i serà bo començar per ella. A l'esquerra repetim aquell exemple (3 valors en només dues llistes i també quatre llistes amb només 3 valors, a la 4ª fila), interpretant ara que la casella 5,9 (**5**) està consolidada, 9,6 és l'emplenament actual i **RE-MEMBER** examina el candidat **5**.

Les últimes modificacions havíen servit perquè el veredicte negatiu passés dels 50 segons d'espera d'abans a ser pràcticament instantani i, posats a verificar això, aprofitàvem per completar la SUDOKU-SOLUCIÓ, no fos cas que pel camí sorgís algun altre contratemps. Val a dir que ho fèiem encreuant els dits, desitjant de tot cor que no passés cap desgràcia... però va passar: dels candidats admesos a 9,6, vam escollir el **4** (si haguéssim triat algun dels altres, **3**, **6**, **7** o **8** no hagués passat res, ves per on) i tot seguit vam anar a emplenar la casella 6,2, on els primers valors compatibles, **1**, **2**, **4** i **5**, van aparèixer immediatament, però després d'això va caldre esperar 1 minut i 20 segons abans no sortís la invitació *CLIC sobre el nombre que va a la casella marcada:* indicant que no hi havia més candidats vàlids. La inviabilitat dels candidats **7** i **8** era trivial, com ho havia estat la de **3**, així que la responsabilitat de la inesperada espera només podia ser de **6** i **9**, i fins i tot l'autor havia percebut una breu interrupció del missatge *Espereu el missatge "CLIC sobre el nombre..."* (que durant l'espera embolcalla el caseller 9×9 central) que podia correspondre a l'arribada del veredicte negatiu sobre el 6, així que els 80 segons d'espera dels candidats **6** i **9** semblaven desglossar-se en 8 segons per al primer i 72 per al segon. Les representacions del mig i de la dreta, mostren una i altra situació, i hem aprofitat (al marge de les ocupacions en negreta que, com de costum, corresponen a les consolidades -tret de la casella actual, subratllada- en el moment de produir-se el conflicte) per incloure-hi l'últim emplenament virtual que precedeix cada veredicte negatiu. Com que la causa que les últimes recursions de **RE-MEMBER** no progressin és representativa del que passa amb les autorecursions precedents, les exposarem ara mateix, cosa que també servirà per avisar d'entrada al lector sobre el significat del caràcter X (que fins ara havíem utilitzat quasi exclusivament per referir-nos a elements **(nil nil nil nil nil nil nil nil nil)** de **(cadr 2L)**) quan el veiem ocupant les caselles lliures que quedin a la quarta fila.

0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0
0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0
0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0
0 0 0	0 0 0	0 0 **5**	0 0 0	0 0 0	0 0 **4**	0 0 0	0 0 0	0 0 **4**
0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0
3 6 7	0 0 0	0 0 0	**3 6 7**	1 8 2	X X 5	**3 6 7**	1 X 2	X X X
0 0 0	0 0 0	0 0 0	0 0 0	5 9 1	0 0 0	0 0 0	5 6 1	0 0 0
0 0 0	0 0 0	0 0 0	9 8 3	2 4 **6**	0 0 0	8 7 6	2 4 **9**	0 0 0
4 5 1	0 0 0	**2 6 9**	**4 5 1**	8 7 3	**2 6 9**	**4 5 1**	8 7 3	**2 6 9**

En la representació del mig ens situarem en l'emplenament virtual de 3,2 (3), que desencadenarà els emplenaments automàtics de 5,2 (4), 5,3 (9), 5,4 (8) i 9,4 (5); després d'això s'emplena 4,2 (2), que desencadenarà la seqüència 6,4 (2), 4,4 (1), 4,3 (5) i 6,3 (1); després d'això, l'activació de **FORA** en detectar **ACTUALITZA-PLUS** una infracció a la **norma 1** en la 4ª fila (malgrat no haver-hi cap casella ocupada amb el valor **4**, les caselles 7,4 i 8,4 tampoc no admeten aquest valor, i aquest és el fet que hem volgut significar amb el caràcter X) avortarà una autorecursió que ja era l'última (no queden valors per assajar en les caselles lliures precedents). En la representació de la dreta ens situarem en l'emplenament virtual de 3,2 (6), que desencadenarà els emplenaments automàtics de 5,2 (4) i 5,3 (6); després d'això s'emplena 4,2 (2), que desencadenarà la seqüència 6,4 (2), 4,4 (1), 4,3 (5) i 6,3

(1), com abans; i com abans, l'activació de **FORA** en detectar **ACTUALITZA-PLUS** una infracció a la **norma 1** en la 4ª fila (no hi ha cap casella ocupada amb un **4**, i les 5,4, 7,4, 8,4 i 9,4 tampoc no admeten aquest valor) avortarà una autorecursió que ja era l'última (no queden valors per assajar en les caselles lliures precedents).

Recuperada l'esma, la primera reacció havia estat suposar que, com tantes altres experiències similars, tot venia de l'avarícia amb què inicialment ens plantejàvem els dispositius d'alleujament de **RE-MEMBER**, restrintgint-los al cas **INI** (casella actual) per por que generalitzant-los a totes les recursions (totes les caselles lliures) milloréssim els temps en uns quants emplenaments a canvi d'empitjorar-los en la resta. Ja havíem hagut de baixar del burro en l'abast de l'acció tapaforats i en l'acció controladora de la **norma 1**. ¿Que postser havia arribat l'hora de fer el mateix amb l'acció controladora de les **normes 2**/**3**, fins ara limitada a **INI**, i estendre-la a totes les iteracions? Tenint clar que aquesta generalització no la podíem escometre així com així, perquè els algorismes de detecció d'infraccions a aquestes normes, aplicats ara a requadres 3×3, files i columnes (fins i tot tenint en compte la millora que suposa que **EXPL-SUBCONJ** treballi amb unes **LLISTES** creades a **INI-LLISTES** una vegada per totes, introduïda al final de l'últim subcapítol per tal de contrarrestar l'ampliació a aquestes últimes unitats) eren massa lents com per actuar en qualsevol circumstància, s'havia arribat a pensar en una solució de compromís que, perquè el lector se'n pugui fer càrrec, ens obliga a obrir un breu parèntesi (un més!). En l'últim capítol, *Consells pràctics per evitar una executio precox*, proposem una possibilitat per al cas que a l'usuari li sigui relativament indiferent emplenar una casella amb un valor o amb un altre: si en el caseller 9×9 ja han aparegut alguns valors i no té paciència per esperar que acabi de sortir la resta, pot fer doble clic sobre algun dels que ja veu en pantalla; el valor queda assignat a la casella actual i la cerca de més candidats s'interromp. Doncs bé: la solució de compromís que hem esmentat consistiria a seguir reservant a la casella actual amb cadascun dels valor candidats (**INI** = **T**) l'accés al dispositiu detector d'infraccions a les **normes 2**/**3**, amb caràcter general, però amb una excepció sota demanda: si quan el veredicte sobre la candidatura d'algun o alguns valors trigués massa i a l'usuari no l'importés haver d'esperar més del compte en el cas concret d'aquella casella, fent doble clic sobre qualsevol de les caselles buides (tant si corresponia a un valor ja refusat com si corresponia a un altre encara pendent de veredicte) podria aconseguir que **RE-MEMBER** interrompés l'exploració del candidat en curs i tornés a realitzar-la (repetint candidat i seguint amb els que quedessin per explorar) però fent intervenir aquell dispositiu... i, naturalment, prenent-se tot el temps que calgués. ¡Menys mal que, a diferència d'altres ocasions, l'autor no es va llançar a passar aquesta ocurrència a codi, perquè hagués estat una total i absoluta pèrdua de temps! Per què diem això? Doncs perquè, per una vegada, se li va ocórrer comprovar si la nova ensopegada obeïa al diagnòstic que li semblava tan evident, i va resultar que no: anul·lant tots els condicionaments a **INI** (tret del de l'accés a **TERCETS** < **EXPLORA-3*9**, perquè ja n'hi hauria hagut massa) la victòria pírrica va consistir a baixar de 80 a 68 segons; això sí, apujant considerablement els temps d'espera a totes les caselles no emplenades automàticament (en començar, els temps es multiplicaven per 10, i en arribar-li el torn a la nostra casella 6,2 ja "només" es multiplicaven per 3).

Davant d'això, hi caben dues actituds: creure resignadament que ja s'ha arribat al límit de les possibilitats d'alleugerir l'acció de **RE-MEMBER** (infal·lible com una piconadora, però que sense additaments és incapaç de mirar més enllà de la primera casella lliure), i consolar-se pensant que casos com aquest no són massa freqüents i que, si ens remetem a les encallades que teníem a l'inici d'aquest macrocapítol, 80 segons són *peccata minuta*; abans de llençar la tovallola, fer un últim intent i estudiar les trajectòries que condueixen a aquesta durada anòmala, per si dessota del caos aparent descobríssim algun indici d'ordre que ens permetés fer drecera. Com que l'autor és cabut com ell sol, ja ho sabeu, sobren prolegòmens i ja podeu armar-vos de paciència per reconstruir els primers passos de l'embranzida fallida de la qual mostràvem la conclusió a la pàgina precedent. Com allà, començarem amb el veredicte menys llarg (mai no direm acceptable, perquè 8 segons no ho són), que correspon al candidat **6**, i seguirem amb el del candidat **9** (recordeu: casella 6,2).

Veureu com limitant-nos als sis primers intents de cada cas (que, qualitativament, ofereixen poques novetats respecte a l'últim) en tindrem prou per apreciar quina és la tònica general, i sobretot per adonar-nos que per determinar l'aptitud dels candidats **6** i **9** no és imprescindible sotmetre'ls al procediment general, perquè ja està implícita d'entrada, des que aquests valors són assignats virtualment a 6,2. Només cal fer una mica de neteja en el contingut de les llistes que componen **(cadr**

2R), anàlogament a com ho feia **FILS/COLS** per explicitar l'estructura dels tercets repetits en requadres 9×3 o 3×9 (configuracions **AAA-AAA** o **AAA-BBB**, que hem tractat en el subcapítol precedent), perquè aflori la contradicció essencial i la partida quedi resolta en tres jugades, evitant que el veredicte estigui condicionat a la conclusió d'un inacabable seguit de giragonses similars a les que ara presentarem.

La primera incursió (1ª renglera, esquerra) arriba sense incidències a la casella 4,2 (1), ocupació virtual que desencadena quatre emplenaments automàtics: 6,4 (1), 4,4 (2), 4,3 (5) i 6,3 (2). Tot seguit, l'emplenament rutinari de 5,2 (4) provoca l'automàtic de 5,3 (9), que interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació (**F-3*3**) ni com a valor potencial (**F-LL3**). No representarem l'últim intent a 5,2 (9), que provoca l'automàtic de 5,3 (4), perquè acaba exactament igual. Anomenarem segona incursió (1ª renglera, centre) la regressió a 4,2 (4), que provoca l'emplenament automàtic de 5,2 (9) i ja no passa d'aquí, en quedar 5,3 sense valors potencials. Anomenarem tercera incursió (1ª renglera, dreta) la regressió a 4,2 (5), seguida per l'emplenament rutinari de 5,2 (9), per l'automàtic de 5,3 (4) i, de nou, pels rutinaris de 7,2 (4), 8,2 (1), 9,2 (7), 1,3 (6), 2,3 (7), 3,3 (9) i 4,3 (1). Tot seguit venen tres emplenaments automàtics, 6,3 (2), 6,4 (1) i 4,4 (2), i després d'això s'interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a valor potencial. No representarem l'últim intent a 4,3 (2), que provoca els emplenaments automàtics 6,3 (1), 4,4 (1) i 6,4 (2), perquè acaba exactament igual.

0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0
0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0
0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0
0 0 0	0 0 0	0 0 **4**	0 0 0	0 0 0	0 0 **4**	0 0 0	0 0 0	0 0 **4**
0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0
3 6 7	2 X 1	X X X	**3 6 7**	0 0 0	0 0 0	**3 6 7**	2 X 1	X X X
0 0 0	5 9 2	0 0 0	0 0 0	0 X 0	0 0 0	6 7 9	1 4 2	0 0 0
2 3 8	1 4 **6**	0 0 0	2 3 8	4 9 **6**	0 0 0	2 3 8	5 9 **6**	4 1 7
4 5 1	3 7 8	**2 6 9**	**4 5 1**	3 7 8	**2 6 9**	**4 5 1**	3 7 8	**2 6 9**

0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0	0 0 0	0 **5** 0	0 0 0
0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0	0 0 0	0 **2** 0	0 0 0
0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0	0 0 0	0 **1** 0	0 0 0
0 0 0	0 0 0	0 0 **4**	0 0 0	0 0 0	0 0 **4**	0 0 0	0 0 0	0 0 **4**
0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0	0 0 0	0 0 0	**1 2** 0
3 6 7	0 0 0	0 0 0	**3 6 7**	2 8 1	X X 5	**3 6 7**	2 8 1	X X 5
0 0 0	0 X 0	0 0 0	6 7 8	5 9 2	0 0 0	6 7 8	1 9 2	0 0 0
2 3 8	9 4 **6**	0 0 0	2 3 9	1 4 **6**	0 0 0	2 3 9	5 4 **6**	7 1 8
4 5 1	3 7 8	**2 6 9**	**4 5 1**	3 7 8	**2 6 9**	**4 5 1**	3 7 8	**2 6 9**

La quarta incursió representada (2ª renglera, esquerra), cinquena en realitat, és l'últim intent a 4,2 (9), que provoca l'emplenament automàtic de 5,2 (4) i acaba com l'antepenúltim (el de 1ª renglera, centre). La cinquena (2ª renglera, centre) obre el segon i últim intent en la posició 3,2 (9), emplenament rutinari que en desencadena quatre d'automàtics: 5,2 (4), 5,3 (9), 5,4 (8) i 9,4 (5). El següent emplenament rutinari, 4,2 (1), en desencadena quatre més d'automàtics: 6,4 (1), 4,4 (2), 4,3 (5) i 6,3 (2). Tot seguit s'interromprà aquesta línia d'acció, per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a valor potencial. La sisena (2ª renglera, dreta) conserva les caselles 5,2 (4), 5,3 (9), 5,4 (8) i 9,4 (5) ocupades automàticament, és l'últim intent a 4,2 (5), seguit pels emplenaments rutinaris de 7,2 (7), 8,2 (1), 9,2 (8), 1,3 (6), 2,3 (7), 3,3 (9) i 4,3 (1). Tot seguit, tres emplenaments automàtics, 6,3 (2), 6,4 (1) i 4,4 (2), i després d'això s'interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a valor potencial. No representarem l'últim intent a 4,3 (2), que provoca els emplenaments automàtics 6,3 (1), 4,4 (1) i 6,4 (2), perquè acaba exactament igual.

Passant del candidat **6** al **9** (6,2), la primera incursió (1ª renglera, esquerra) arriba sense incidències a la casella 3,2 (6), ocupació virtual que desencadena l'automàtica de 5,2 (4) i 5,3 (6). Tot seguit, l'emplenament rutinari de 4,2 (1) en desencadena quatre d'automàtics: 6,4 (1), 4,4 (2), 4,3 (5) i 6,3 (2), cosa que interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a valor potencial. La segona incursió (1ª renglera, centre) conserva les caselles 5,2 (4) i 5,3 (6) emplenades automàticament i és l'últim intent a la casella 4,2 (5), seguit pels emplenaments rutinaris de 7,2 (7), 8,2 (1), 9,2 (8), 1,3 (7), 2,3 (8), 3,3 (9) i 4,3 (1). Tot seguit, tres emplenaments automàtics, 6,3 (2), 6,4 (1) i 4,4 (2), i després d'això s'interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a valor potencial. No es representa l'últim intent a 4,3 (2), que provoca els emplenaments automàtics 6,3 (1), 4,4 (1) i 6,4 (2), perquè acaba exactament igual. Anomenarem tercera incursió (1ª renglera, dreta) la que obre el segon i últim intent en la posició 3,2 (8), i que segueix amb l'emplenament rutinari de 4,2 (1). Aquesta ocupació en desencadena quatre d'automàtiques: 6,4 (1), 4,4 (2), 4,3 (5) i 6,3 (2). El següent emplenament rutinari, 5,2 (4), en desencadena un d'automàtic, 5,3 (6), amb què s'interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a valor potencial. No representarem el segon i últim intent a 5,2 (6), que provoca l'emplenaments automàtic de 5,3 (4), perquè acaba exactament igual.

0 0 0 0 0 0 0 0 0	0 **5** 0 0 **2** 0 0 **1** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **3 6 7**	0 0 0 0 0 0 2 X 1	0 0 **4** **1 2** 0 X X X
0 0 0 2 3 6 **4 5 1**	5 6 2 1 4 **9** 3 7 8	0 0 0 0 0 0 **2 6 9**

0 0 0 0 0 0 0 0 0	0 **5** 0 0 **2** 0 0 **1** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **3 6 7**	0 0 0 0 0 0 2 X 1	0 0 **4** **1 2** 0 X X X
7 8 9 2 3 6 **4 5 1**	1 6 2 5 4 **9** 3 7 8	0 0 0 7 1 8 **2 6 9**

0 0 0 0 0 0 0 0 0	0 **5** 0 0 **2** 0 0 **1** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **3 6 7**	0 0 0 0 0 0 2 X 1	0 0 **4** **1 2** 0 X X X
0 0 0 2 3 8 **4 5 1**	5 6 2 1 4 **9** 3 7 8	0 0 0 0 0 0 **2 6 9**

0 0 0 0 0 0 0 0 0	0 **5** 0 0 **2** 0 0 **1** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **3 6 7**	0 0 0 0 0 0 0 0 0	0 0 **4** **1 2** 0 **3 6 7**
0 0 0 2 3 8 **4 5 1**	0 X 0 4 6 **9** 3 7 8	0 0 0 0 0 0 **2 6 9**

0 0 0 0 0 0 0 0 0	0 **5** 0 0 **2** 0 0 **1** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **3 6 7**	0 0 0 0 0 0 2 X 1	0 0 **4** **1 2** 0 X X X
6 7 9 2 3 8 **4 5 1**	1 4 2 5 6 **9** 3 7 8	0 0 0 1 4 7 **2 6 9**

0 0 0 0 0 0 0 0 0	0 **5** 0 0 **2** 0 0 **1** 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 **3 6 7**	0 0 0 0 0 0 0 0 0	0 0 **4** **1 2** 0 **3 6 7**
0 0 0 2 3 8 **4 5 1**	0 X 0 6 4 **9** 3 7 8	0 0 0 0 0 0 **2 6 9**

La quarta incursió representada (2ª renglera, esquerra), cinquena en realitat, és una regressió a 4,2 (4), que provoca l'emplenament automàtic de 5,2 (6) i ja no passa d'aquí, en quedar 5,3 sense valors potencials. i acaba com l'antepenúltim (el de 1ª renglera, centre). La cinquena (2ª renglera, centre) torna a 4,2 (5) i segueix amb l'emplenament rutinari de 5,2 (6), per l'automàtic de 5,3 (4) i, de nou, pels rutinaris de 7,2 (1), 8,2 (4), 9,2 (7), 1,3 (6), 2,3 (7), 3,3 (9) i 4,3 (1). Tot seguit venen tres emplenaments automàtics, 6,3 (2), 6,4 (1) i 4,4 (2), i després d'això s'interromprà aquesta línia d'acció per incompliment de la **norma 1** a la 4ª fila: el 4 no hi és present, ni com a valor d'ocupació ni com a potencial (ens hem saltat l'intent 5,2 (6) perquè, després de l'emplenament automàtic de 5,3 (6), no aconseguiria acabar la 2ª fila). La sisena incursió (2ª renglera, dreta) és l'últim intent a 4,2 (6), que provoca l'emplenament automàtic de 5,2 (4) i que acaba com la penúltima.

Si classifiquem les progressions frustrades en el primer requadre 9×3 (les que hem vist es circumscriuen en aquest àmbit, i seria fàcil d'inferir que les intermèdies també ho faran) en llargues (la infracció de la **norma 1** a la 4ª fila s'esdevé quan els emplenaments ordinaris -automàtics a part- han arribat a la 3ª fila), mitjanes (s'esdevé quan els emplenaments han arribat a 2ª fila) i curtes (la interrupció es

deu al fet que la casella 5,3 s'ha quedat sense poder acceptar cap valor 1... 9), els dos septets d'intents considerats (els sis inicials i l'últim, amb els valors candidats **6** i **9** a 6,2) es comporten de forma molt semblant: dues de llargues, tres de mitjanes i dues de curtes. La causa d'interrupció predominant (5 casos sobre 2) és la infracció de la **norma 1** a la 4ª fila, per incompareixença del valor **4**, sigui com a valor real o potencial. Curiosament, les combinacions de valors possibles, pel que fa a les 6 primeres caselles lliures (tercets 4,1-5,1-6,1 i 1,2-2,2-3,2), també és de 192 en ambdós casos, producte de les 6 permutacions de 3 valors en el primer (3-7-8, 3-8-7, 7-3-8, 7-8-3, 8-3-7 i 8-7-3) per les 32 variacions que admet el segon (2-3-8, 2-3-9, 2-7-3, 2-7-8, 2-7-9, 2-8-3, 2-8-9, 2-9-3, 2-9-8, 7-2-3, 7-2-8, 7-2-9, 7-3-2, 7-3-8, 7-3-9, 7-8-2, 7-8-3, 7-8-9, 7-9-2, 7-9-3, 7-9-8, 8-2-3, 8-2-9, 8-3-2, 8-3-9, 8-7-2, 8-7-3, 8-7-9, 8-9-2, 8-9-3, 9-8-2 i 9-8-3, amb el **6** a 6,2, i 2-3-6, 2-3-8, 2-7-3, 2-7-6, 2-7-8, 2-8-3, 2-8-6, 6-2-3, 6-2-8, 6-3-2, 6,3-8, 6-7-2, 6-7-3, 6-7-8, 6-8-2, 6-8-3, 7-2-3, 7-2-6, 7-2-8, 7-3-2, 7-3-6, 7-3-8, 7-8-2, 7-8-3, 7-8-6, 8-2-3, 8-2-6, 8-3-2, 8-3-6, 8-7-2, 8-7-3 i 8-7-6, amb el **9** a 6,2), cosa que no contribueix a explicar que en el segon cas la durada total de l'exploració sigui 9 vegades superior a la del primer: només podem suposar que, fora de les incursions contemplades (les 6 primeres i l'última), les freqüències de les curtes, mitjanes i llargues difereixen considerablement entre una i altra exploració (amb una incidència més gran de les curtes amb el candidat **6** i de les llargues amb el candidat **9**, probablement). En qualsevol cas, la longitud de les trajectòries **6** i **9** sí que justifica que l'aparició del veredicte (o no aparició, per ser més precisos) no sigui pràcticament instantània.

Un cop vist que les raons de no acceptació d'aquests candidats es focalitzen en la 4ª fila, interessaria veure l'aspecte de **(cadr 2R)** o **R-LLL** (que, no havent-hi cap configuració **AAA-AAA** o **AAA-BBB**, són el mateix) just acabats d'introduir (**INI** = **T**), per si podem percebre-hi alguna singularitat. Com que la representació d'aquestes pseudomatrius 9×9 de **nil**s i llistes de 9 valors és problemàtica, en el context del format de text que utilitzem (normalment ens limitem a representar el fragment que correspon a un requadre 3×3, i quan hem hagut de representar el corresponent a una fila sencera hem recorregut a diversos subterfugis), ens limitarem a reproduir-ne una part: el conjunt format pel requadre inferior-centre, central (centre-centre) i la 1ª fila del requadre centre-dreta, de manera que també hi surti representada la part no ocupada de la conflictiva 4ª fila. Tot i amb aquesta restricció, ens ha calgut recórrer a caràcters comprimits horitzontalment i, per suposat, seguir amb els zeros en substitució dels valors **nil**.

Començarem pel candidat **6** a la posició 6,2. Com era de preveure, les **normes 1**, **2** i **3** es compleixen escrupolosament en tots els requadres 3×3, files i columnes, i, en particular, en la 4ª fila. Però si haguéssim de fer algun comentari, el faríem a propòsit del requadre inferior-centre i, en concret, del seu tercet inferior: si els valors **3**, **7** i **8** (els hem subratllat) hi són obligats (són únics en el tercet), ¿per què encara apareixen, com a valors potencials, en els sis tercets restants? Doncs simplement perquè la funció **ACT-LLL** < **ACTUALITZA** no contemplava cap actuació depuradora en aquest sentit, ni tampoc se'ns va ocórrer de fer-ho a **TERCETS** (més ben dit, sí que **TERCETS** ho detectava i abordava la depuració, però si finalment **FILS/COLS** no confirmava l'existència de configuracions **AAA-AAA** o **AAA-BBB**, l'acció quedava sense efecte).

```
(1 2 3 0 5 6 7 8 9) (0 0 3 0 0 6 7 8 9) (1 2 3 4 5 0 7 8 9)
(0 0 3 4 5 6 7 8 9) (0 0 3 4 0 6 7 8 9) (0 0 3 4 5 0 7 8 9)
(1 2 0 4 5 0 0 8 9) (0 0 0 4 0 0 0 8 9) (1 2 0 4 5 0 0 8 9) (0 0 0 0 5 0 0 8 9) (0 0 0 0 5 0 0 8 9) (0 0 0 0 5 0 0 8 0)

(1 2 3 4 5 0 7 8 9) (0 0 3 4 0 0 7 8 9) (1 2 3 4 5 0 7 8 9)
(1 2 3 4 5 0 7 8 9) (0 0 3 4 0 0 7 8 9)         0
(0 0 3 0 0 0 7 8 0) (0 0 3 0 0 0 7 8 0) (0 0 3 0 0 0 7 8 0)
```

Prenem-ne nota, suposem que en un lloc o altre podríem materialitzar la depuració esmentada, de moment incorporem-la manualment a la representació, i seguim. Perquè ara, en aclarir-se el panorama en el requadre inferior-centre i, en concret, el de les caselles 5,2 i 5,3, podríem seguir amb un altre pregunta: si els valors **4** i **9** (els hem subratllat) hi són obligats (són únics en aquest parell de caselles, que pertanyen als mateixos requadre i columna), ¿per què encara apareixen com a valors potencials en aquest requadre 3×3 i a la 5ª columna? Ara veiem que la cosa va més enllà de la problemàtica dels tercets (de fet, **TERCETS** només intervé quan les tres caselles amb tres valors potencials exclusius estan alineades, però la repercusió depuradora en el requadre subsisteix encara que no ho estiguin).

(**1 2 3** 0 **5 6 7 8 9**) (0 0 **3** 0 0 **6 7 8 9**) (**1 2 3 4 5** 0 **7 8 9**)
(0 0 **3 4 5 6 7 8 9**) (0 0 **3 4** 0 **6 7 8 9**) (0 0 **3 4 5** 0 **7 8 9**)
(**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 **4** 0 0 0 **8 9**) (**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8** 0)

(**1 2** 0 **0 5** 0 0 0 **0**) (0 0 0 **4** 0 0 0 0 **9**) (**1 2 3 0 5** 0 **7 8 0**)
(**1 2** 0 **0 5** 0 0 0 **0**) (0 0 0 **4** 0 0 0 0 **9**) 0
(0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0)

Farem com abans i abordarem manualment la depuració, i és ara quan ens adonem que a la 4ª fila hi ha 3 valors (**1**, **2** i **4**) en només 2 llistes (4,4 i 6,4) i 4 llistes (5,4, 7,4, 8,4 i 9,4) amb només 3 valors (**5**, **8** i **9**), és a dir, una clara infracció de les **normes 2**/**3**:

(**1 2 3** 0 **5 6 7 8 9**) (0 0 **3** 0 0 **6 7 8** 0) (**1 2 3 4 5** 0 **7 8 9**)
(0 0 **3 4 5 6 7 8 9**) (0 0 **3** 0 0 **6 7 8** 0) (0 0 **3 4 5** 0 **7 8 9**)
(**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 **0** 0 0 0 **8** 0) (**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8** 0)

(**1 2** 0 **0 5** 0 0 0 **0**) (0 0 0 **4** 0 0 0 0 **9**) (**1 2 3 0 5** 0 **7 8 0**)
(**1 2** 0 **0 5** 0 0 0 **0**) (0 0 0 **4** 0 0 0 0 **9**) 0
(0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0)

Això ho hem descobert nosaltres però, encara que **ACTUALITZA-PLUS** tingués accés a aquesta versió depurada de **R-LLL**, ni ho arribaria a ensumar perquè, atès l'ordre en què hem situat els diversos dispositius, abans s'emplenarien automàticament 5,4 (**8**), 9,4 (**5**) i 7,4 (**9**):

(**1 2 3** 0 **5 6 7** 0 **9**) (0 0 **3** 0 0 **6 7** 0 0) (**1 2 3 4 5** 0 **7** 0 **9**)
(0 0 **3 4 5 6 7** 0 **9**) (0 0 **3** 0 0 **6 7** 0 0) (0 0 **3 4 5** 0 **7** 0 **9**)
(**1 2** 0 **4** 0 0 0 0 0) 0 (**1 2** 0 **4** 0 0 0 0 0) 0 (0 0 0 0 0 0 0 0 0) 0

(**1 2** 0 **4 5** 0 0 0 **9**) (0 0 0 **4** 0 0 0 0 **9**) (**1 2 3 4 5** 0 **7 8 9**)
(**1 2** 0 **4 5** 0 0 0 **9**) (0 0 0 **4** 0 0 0 0 **9**) 0
(0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7** 0 0) (0 0 **3** 0 0 0 **7 8** 0)

Destaquem que, després del segon forat tapat, 9,4 (**5**), tant 7,4 com 8,4 són forats candidats a l'assignació automàtica del **9**, però **ACTUALITZA-PLUS** localitzaria abans el primer, que seria l'agraciat. Després d'això, el segon forat ja no admetria cap més valor i seria la llista **(nil nil nil nil nil nil nil nil nil)** corresponent, que expressa aquest fet, la que provocaria l'activació de **FORA**, la interrupció de l'exploració amb **6** i el refús d'aquest candidat a la casella 6,2.

El refús del candidat **9** prescindirà d'aquests últims tràmits però igualment caldrà fer neteja a **R-LLL** perquè la situació contradictòria quedi al descobert. Igual que abans, les **normes 1**, **2** i **3** es compleixen escrupolosament en tots els requadres, files i columnes, i, en particular, en la 4ª fila, i, si haguéssim de fer algun comentari, el faríem a propòsit del requadre inferior-centre i, en concret, del seu tercet inferior: si els valors **3**, **7** i **8** (els hem subratllat) hi són obligats (són únics en el tercet), ¿per què encara apareixen, com a valors potencials, en els sis tercets restants?

(**1 2 3** 0 **5 6 7 8 9**) (0 0 **3** 0 0 **6 7 8 9**) (**1 2 3 4 5 6 7 8** 0)
(0 0 **3 4 5 6 7 8 9**) (0 0 **3 4** 0 **6 7 8 9**) (0 0 **3 4 5 6 7 8** 0)
(**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 **4** 0 0 0 **8 9**) (**1 2** 0 **4 5** 0 0 **8** 0) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 0**)

(**1 2 3 4 5 6 7 8 9**) (0 0 **3 4** 0 **6 7 8** 0) (**1 2 3 4 5 6 7 8** 0)
(**1 2 3 4 5 6 7 8 9**) (0 0 **3 4** 0 **6 7 8** 0) 0
(0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0)

Després d'efectuar manualment la depuració, s'aclareix el panorama en el requadre inferior-centre i, en concret, el de les caselles 5,2 i 5,3: si els valors **4** i **6** (els hem subratllat) hi són obligats (són únics en aquest parell de caselles, que pertanyen als mateixos requadre i columna), ¿per què encara apareixen com a valors potencials en aquest requadre 3×3 i a la 5ª columna?

(**1 2 3** 0 **5 6 7 8 9**) (0 0 **3** 0 0 **6 7 8 9**) (**1 2 3 4 5 6 7 8** 0)
(0 0 **3 4 5 6 7 8 9**) (0 0 **3 4** 0 **6 7 8 9**) (0 0 **3 4 5 6 7 8** 0)
(**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 **4** 0 0 0 **8 9**) (**1 2** 0 **4 5** 0 0 **8** 0) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8** 0)

(**1 2** 0 **4 5 6** 0 0 **9**) (0 0 0 **4** 0 **6** 0 0 0) (**1 2** 0 **4 5 6** 0 0 0)
(**1 2** 0 **4 5 6** 0 0 **9**) (0 0 0 **4** 0 **6** 0 0 0) 0
(0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0)

Farem com abans i abordarem manualment la depuració, i és ara quan ens adonem que a la 4ª fila hi ha 3 valors (**1, 2** i **4**) en només 2 llistes (4,4 i 6,4) i 4 llistes (5,4, 7,4, 8,4 i 9,4) amb només 3 valors (**5, 8** i **9**), és a dir, una clara infracció de les **normes 2/3.** A diferència del cas precedent (candidat **6**), no hi ha cap forat per tapar, així que **ACTUALITZA-PLUS** la localitzarà i activarà **FORA**, provocant la interrupció de l'exploració amb **9** i el refús d'aquest candidat a la casella 6,2:

(**1 2 3** 0 **5 6 7 8 9**) (0 0 **3** 0 0 0 **7 8 9**) (**1 2 3 4 5 6 7 8** 0)
(0 0 **3 4 5 6 7 8 9**) (0 0 **3** 0 0 0 **7 8 9**) (0 0 **3 4 5 6 7 8** 0)
(**1 2** 0 **4 5** 0 0 **8 9**) (0 0 0 0 0 0 0 **8 9**) (**1 2** 0 **4 5** 0 0 **8** 0) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8 9**) (0 0 0 0 **5** 0 0 **8** 0)

(**1 2** 0 0 **5** 0 0 0 **9**) (0 0 0 **4** 0 **6** 0 0 0) (**1 2** 0 0 **5** 0 0 0 0)
(**1 2** 0 0 **5** 0 0 0 **9**) (0 0 0 **4** 0 **6** 0 0 0) 0
(0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0) (0 0 **3** 0 0 0 **7 8** 0)

Veient per on van els trets, no caldrà esplaiar-se tant per descriure la segona ensopegada, que va consistir en l'absoluta inutilitat de l'última versió substituint la precedent: buscant un exemple d'infracció de les **normes 2/3** a la 8ª fila, amb set valors (**1, 2, 3, 4, 5, 6** i **7**) en només sis llistes i tres llistes amb només 2 valors (**8** i **9**), que ara repetim a la dreta i que convé omplir deixant per al final les caselles 2,7 (**7**) i 5,7 (**1**), no només havíem fracassat en la cerca sinó que, en arribar a l'ultima d'aquestes caselles, el veredicte (negatiu) trigava més del que fóra raonable; el més decebedor era que, a diferència dels casos on la norma es respectava a nivell de requadres 3×3 però no en files o columnes, l'aplicació de l'última versió no servia per evitar la demora, perquè el problema no era exactament el mateix.

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
0 **7** 0	0 **1** 0	0 0 0
0 **6** 0	0 **7** 0	0 **1** 0
0 **5** 0	0 **6** 0	0 **7** 0
0 **4** 0	0 **5** 0	0 **6** 0
0 **3** 0	0 **4** 0	0 **5** 0
0 **2** 0	0 **3** 0	0 **4** 0
0 **1** 0	0 **2** 0	0 **3** 0

Ja llavors avançàvem que l'arrel del conflicte podia ser una infracció que estava implícita en l'estat d'emplenament però que encara no es reflectia a **R-LLL** i, per tant, no es detectava a temps d'impedir una exploració intrincada i llarguíssima. Ara sabem que cal detectar prèviament aquestes situacions singulars per actuar en conseqüència, simplificant les llistes de **R-LLL** anàlogament a com ho fèiem amb el dispositiu **TERCETS**, per tal que la supervisió de les **normes 2/3** dispari l'alarma.

Com a l'exemple anterior, en treurem l'entrellat recorrent a la representació dels fragments de **R-LLL** implicats, però ara la necessitat de presentar el requadre 9×3 superior tot sencer encara accentua els problemes d'espai, cosa que ens obligarà a simplificar l'aspecte de les llistes no directament implicades en el problema (1ª, 3ª, 4ª, 6ª, 7ª i 9ª de cada fila). Els altres requadres 9×3 s'ometen perquè només ens podria interessar la 8ª columna de caselles, però com que estan ocupades, tret de les representades a nivell de **R-LLL**, ja en tindrem prou per apreciar què passa.

(**1** ... **9**) (0 0 0 0 0 0 0 **8 9**) (**1** ... **9**) (0 ... **9**) (0 0 0 0 0 0 0 **8 9**) (0 ... **9**) (**1** ... **9**) (0 **2** 0 0 0 0 0 **8 9**) (**1** ... 9)
(**1** ... **9**) (0 0 0 0 0 0 0 **8 9**) (**1** ... **9**) (0 ... **9**) (0 0 0 0 0 0 0 **8 9**) (0 ... **9**) (**1** ... **9**) (0 **2** 0 0 0 0 0 **8 9**) (**1** ... 9)
(**1** ... **9**) 0 (**1** ... **9**) (0 ... **9**) 0 (0 ... **9**) (**1** ... **9**) (0 **2** 0 0 0 0 0 **8 9**) (**1** ... 9)

Hem subratllat els únics valors no nuls de les llistes corresponents al parell de caselles 2,8 i 5,8 (8ª fila) i al parell 2,9 i 5,9 (9ª fila), **8** i **9**. Cada parell escombrarà aquests valors de les llistes de la seva fila, per innecessaris, cosa que apreciem en la representació que ve a continuació:

(**1** ... 0) (0 0 0 0 0 0 0 **8 9**) (**1** ... 0) (0 ... 0) (0 0 0 0 0 0 0 **8 9**) (0 ... 0) (**1** ... 0) (0 **2** 0 0 0 0 0 0 0) (**1** ... 0)
(**1** ... 0) (0 0 0 0 0 0 0 **8 9**) (**1** ... 0) (0 ... 0) (0 0 0 0 0 0 0 **8 9**) (0 ... 0) (**1** ... 0) (0 **2** 0 0 0 0 0 0 0) (**1** ... 0)
(**1** ... **9**) 0 (**1** ... **9**) (0 ... **9**) 0 (0 ... **9**) (**1** ... **9**) (0 **2** 0 0 0 0 0 **8 9**) (**1** ... 9)

Aquí és on es fan palesos els efectes sobre la 8ª columna (reduïda a les posicions 8,7, 8,8 i 8,9): infracció de les **normes 2/3**, per tenir 2 valors (**8** i **9**) en només 1 llista (8,7) i 2 llistes (8,8 i 8,9) amb només 1 valor (**2**). Una altra cosa és que al final del subcapítol precedent haguéssim decidit ignorar aquesta tipologia d'infraccions perquè, igual com passava amb el candidat **6** a la casella 6,2 del cas

anterior, entre **ACTUALITZA-PLUS** (tapant amb un **2** el forat 8,8, que és el primer a ser localitzat) i el dispositiu detector de **(nil nil nil nil nil nil nil nil nil)** (que en això s'ha convertit la llista 8,9, com veieu tot seguit) que activa **FORA**, l'exploració promoguda pel candidat **1** a 5,7 s'interromprà, comentari amb què donem per tancada l'anàlisi dels fracassos que havíen patit i la forma de reconduir-los:

(**1** ... 0) (0 0 0 0 0 0 0 **8 9**) (**1** ... 0) (0 ... 0) (0 0 0 0 0 0 0 **8 9**) (0 ... 0) (**1** ... 0) (0 0 0 0 0 0 0 0 **0**) (**1** ... 0)
(**1** ... 0) (0 0 0 0 0 0 0 **8 9**) (**1** ... 0) (0 ... 0) (0 0 0 0 0 0 0 **8 9**) (0 ... 0) (**1** ... 0) (0 **2** 0 0 0 0 0 0 0) (**1** ... 0)
(**1** ... **9**) 0 (**1** ... **9**) (0 ... **9**) 0 (0 ... **9**) (**1** ... **9**) (0 0 0 0 0 0 0 **8 9**) (**1** ... 9)

Tenint present que, en una unitat modular (requadre 3×3, fila o columna) de **R-LLL** només té sentit buscar **n** llistes iguals amb només **n** valors no nuls si hi ha **n+2** caselles lliures (perquè si només en tenim **n**, encara que localitzem les **n** llistes no queda res per depurar, i si en tenim **n+1** pot passar que a la **(n+1)**-èsima hi hagi el **(n+1)**-èsim valor -i en aquest cas **ACTUALITZA-PLUS** ja hauria ocupat la casella- o que no hi sigui -i en aquest cas **ACTUALITZA-PLUS** hauria detectat una infracció de la **norma 1**-), ja queda prou clar què hi hauríem de buscar:
- En unitats amb totes les caselles lliures, cal mirar si hi ha 7 llistes iguals, amb només 7 valors no nuls.
- En unitats a partir de 8 caselles lliures, cal mirar si hi ha 6 llistes iguals, amb només 6 valors no nuls.
- En unitats a partir de 7 caselles lliures, cal mirar si hi ha 5 llistes iguals, amb només 5 valors no nuls.
- En unitats a partir de 6 caselles lliures, cal mirar si hi ha 4 llistes iguals, amb només 4 valors no nuls.
- En unitats a partir de 5 caselles lliures, cal mirar si hi ha 3 llistes iguals, amb només 3 valors no nuls.
- En unitats a partir de 4 caselles lliures, cal mirar si hi ha 2 llistes iguals, amb només 2 valors no nuls.

La cerca s'ha d'estendre a tots els requadres 3×3, files i columnes. Si comencem pels requadres, haurem de comprovar si les caselles implicades en una detecció estan alineades horitzontalment o vertical, per saber si la depuració també cal estendre-la directament a alguna fila o columna (abans que els arribi el torn a aquestes unitats). Si comencem per files i columnes, caldrà comprovar si totes les caselles implicadesen una detecció se situen en el mateix requadre, per saber si la depuració també s'hi ha d'estendre directament.

Cal desterrar d'entrada dos plantejaments erronis que, en l'afany de simplificar el procés, podríem adoptar acríticament:
- Creure que la depuració es pot dur fins a les últimes conseqüències en una única iteració, si l'ordre de cerca és precisament el presentat (començar a buscar les files més poblades i acabar amb les de 2 valors, perquè en els dos primers casos estudiats era la descoberta de 3 llistes iguals amb 3 valors allò que permetia depurar la unitat de manera que l'existència de 2 llistes iguals amb 2 valors quedés en evidència i permetés abordar la segona depuració). Disortadament una flor no fa estiu, i no hi haurà més remei que seguir una dinàmica similar a la d'**ACTUALITZA-PLUS** < **ACTUALITZA** en funció de la variable **PLUS**, iterant fins que deixi de produir-se algun dels supòsits esmentats.
- Practicar una retallada en el llistat anterior, pensant que amb cercar fins a conjunts de 5 llistes iguals amb només 5 valors no nuls ja n'hi ha prou, ometent la cerca de conjunts de 6 llistes iguals amb només 6 valors i de conjunts de 7 llistes iguals amb només 7 valors. Si ens molestem a comentar un despropòsit tan manifest és perquè jugar amb dos errors alhora, que a sobre són contradictoris entre ells, de vegades pot produir la il·lusió d'estar en el bon camí. El d'ara consistiria a combinar una de les premises de l'error a què ens referíem en el punt precedent (que, si la depuració derivada de la detecció d'un conjunt de **m** llistes iguals amb només **m** valors dóna lloc a la detecció d'un altre conjunt de **n** llistes amb només **n** valors, ha de ser **n < m**) amb una presumpció molt més *naïf*: que, si la depuració que es pot derivar de la detecció d'un conjunt de **m** llistes iguals amb només **m** valors dóna lloc a la detecció d'un conjunt de **n** llistes amb només **n** valors, n'hi haurà prou a detectar el segon conjunt per identificar el primer, com ignorant que l'anomenada depuració no és la transformació identitat. La delirant línia argumental podria esqematitzar-se així:
  - Si la depuració que permet l'existència d'un conjunt de 3 llistes amb 3 valors no nuls pot deixar al descobert un conjunt de 2 llistes amb 2 valors no nuls.
  - Si la depuració que permet l'existència d'un conjunt de 4 llistes amb 4 valors no nuls pot deixar al descobert un conjunt de 2 o 3 llistes amb 2 o 3 valors no nuls.

- Si la depuració que permet l'existència d'un conjunt de 5 llistes amb 5 valors no nuls pot deixar al descobert un conjunt de 2, 3 o 4 llistes amb 2, 3 o 4 valors no nuls.
- Si la depuració que permet l'existència d'un conjunt de 6 llistes amb 6 valors no nuls pot deixar al descobert un conjunt de 2 o 3 llistes amb 2o 3 valors no nuls.
- Si la depuració que permet l'existència d'un conjunt de 7 llistes amb 7 valors no nuls pot deixar al descobert un conjunt de 2 llistes amb 2 valors no nuls.

Llavors (formulació recíproca):

- La depuració que permet l'existència d'un conjunt de 2 llistes amb 2 valors no nuls pot deixar al descobert conjunts de 3, 4, 5, 6 o 7 llistes amb 3, 4, 5, 6 o 7 valors no nuls.
- La depuració que permet l'existència d'un conjunt de 3 llistes amb 3 valors no nuls pot deixar al descobert conjunts de 4, 5 o 6 llistes amb 4, 5 o 6 valors no nuls.
- La depuració que permet l'existència d'un conjunt de 2 llistes amb 2 valors no nuls pot deixar al descobert conjunts de 4 llistes amb 4 valors no nuls.

Amb la qual cosa n'hi hauria prou a rastrejar l'existència de conjunts de 3, 4 o 5 llistes amb només 3, 4 o 5 valors no nuls, per dur la depuració fins al final. Algú preguntarà: per què l'autor perd tant de temps parlant d'un disbarat que a ningú no se li escapa que és un disbarat? La resposta hauria de ser una impúdica exhibició de complaença autoflagel·lant: perquè a l'autor sí que li havia passat pel cap l'absurd, i el pòsit judeocristià que habita el seu subconscient encara exigeix una confessió pública del pecat per poder reconciliar-se amb ell mateix.

Tanmateix, si seguíssim considerant que la depuració d'unitats modulars únicament es pot abordar en presència de **n** llistes iguals amb només **n** valors no nuls sí que cometríem un greu error: no només perquè ens hauríem buscat més feina (l'autor en dóna fe perquè tampoc no se n'ha estat d'ensopegar en aquesta pedra, però aquesta vegada es compadirà dels lectors i evitarà castigar-los amb el codi corresponent) sinó perquè hauríem desaprofitat una munió de casos que excedeixen aquests límits tan estrets i malgrat tot admeten una depuració. N'hi haurà prou a prescindir de l'adjectiu subratllat i considerar qualsevol subconjunt de **n** llistes amb només **n** valors no nuls, en el sentit que donàvem a "**n** valors" quan parlàvem de la **norma 2**: nombre de valors diferents i no nuls que apareixen en el conjunt de les **n** llistes. Així, il·lustrant-ho amb el cas de 3 llistes que en conjunt contenen 3 valors, per exemple **7**, **8** i **9**, no ens limitarem a depurar les unitats que tinguin

(0 0 0 0 0 0 **7 8 9**) (0 0 0 0 0 0 **7 8 9**) (0 0 0 0 0 0 **7 8 9**)

sinó també les que tinguin

(0 0 0 0 0 0 0 **8 9**) (0 0 0 0 0 0 **7 8 9**) (0 0 0 0 0 0 **7 8 9**)
(0 0 0 0 0 0 0 **8 9**) (0 0 0 0 0 0 **7** 0 **9**) (0 0 0 0 0 0 **7 8 9**)
(0 0 0 0 0 0 0 **8 9**) (0 0 0 0 0 0 **7** 0 **9**) (0 0 0 0 0 0 **7 8** 0)

Les raons són òbvies: si en el primer cas no hi pot haver a cap altra llista de la mateixa unitat cap d'aquests tres valors (perquè si n'hi hagués un i l'assignéssim a la casella corresponent, el **7** per exemple, les nostres tres llistes es quedarien només amb els valors **8** i **9**, infringint la **norma 2**), el mateix passa en segon cas; i si suposem que, en el tercer o quart cas, aquest valor **7** és admissible en alguna altra posició de la mateixa fila, columna i/o requadre 3×3 (o que els valors **8** o **9** ho són en el segon, tercer o quart cas), en assignar-lo a la casella corresponent, dues de les nostres llistes es quedarien amb un únic valor (que seria assignat de forma automàtica) i la tercera es quedaria buida; tots aquests supòsits conduirien a un veredicte negatiu per al candidat que hagués provocat aquest estat de coses.

Acotades del tot les situacions que cal detectar, no queden dubtes sobre on convé situar el dispositiu que se n'ha de fer càrrec: a **EXPL-SUBCONJ**, funció que va ser especialment pensada per explorar totes les combinacions possibles amb les llistes **R-LLL** corresponents a cada unitat modular i analitzar-ne el contingut. Només hi ha un petit detall que ens obliga a complicar-la una mica i a retocar **INI-LLISTES** per segona vegada: si, cap al final del subcapítol precedent havíem resolt eliminar la subllista inicial de tots els elements de **LLISTES** (les combinacions binàries), en considerar que les infraccions potencials de les **normes 2**/**3** havien estat apartades prèviament, ara les haurem de recuperar per abordar les depuracions (2 llistes amb 2 valors no s'han de desestimar), però a l'hora d'explorar aquestes subllistes ens podrem saltar les combinacions $(n-2)$-èsimes (el control d'infracció de les **normes** arribava fins a les $(n-1)$-èsimes), com ja havíem subratllat a la pàgina precedent. De seguida veureu que la nova versió d'**INI-LLISTES** únicament difereix de l'última en haver substituït **(cdr LLL)** per **LLL** a la penúltima línia. Pel que fa a la funció **EXPL-SUBCONJ**, el desfasament d'una posició entre els intèrvals en què han d'actuar

el dos dispositius detectors (el de llistes tancades de valors admissibles i el d'infraccions de les **normes 2**/**3**), en relació als diversos ordres combinatoris que té cada element de **LLISTES**, es soluciona amb un inevitable desdoblament operatiu (per evitar repeticions, l'obtenció del conjunt de valors diferents presents en el conjunt de les **K** llistes va a càrrec de la nova funció **PRESENTS**) en què les seves activacions estan controlades per les condicions **(< (1+ K) (length F-L))**, com a límit superior per a la primera, i **(> K 2)** com a límit inferior per a la segona.

Quan detectem un conjunt de caselles tancat a efectes de valors admissibles, el guardarem a **P-DEP** i recorrerem a **PLUS**, que és la mateixa variable que servia per forçar el retorn a **ACTUALITZA-PLUS** a la cerca de més incidències, quan s'havien produït assignacions automàtiques, raó per la qual ara caldrà introduir un element diferenciador d'ambdues situacions: aquest paper va a càrrec de la variable **L-DEP**, llista que conté els **K** valors presents a les **K** llistes detectades i que haurem de depurar de la resta de llistes de la unitat modular, i el testimoni que algun cop s'ha hagut de crear **L-DEP** (aquesta variable s'ha d'inicialitzar a **nil** després de cada intervenció depuradora) serà **DEP**. Si el tipus d'unitat on hem localitzat el conjunt tancat de caselles és una fila o una columna, a la nova funció **ACT-DEPURA** (que actualitza **R-LLL** eliminant els valors admissibles obsolets) no li caldrà cap informació específica, perquè aquesta circumstància es immediatament deduïble de **P-DEP**, però quan sigui un requadre 3×3 i les caselles implicades no s'arrenglerin en files o columnes (supòsit en què haurien estat processades prèviament), no li anirà malament un cop de mà: la variable **XY**, local a **ACTUALITZA** i que representa la posició central del requadre, ens indicarà que el conjunt tancat de caselles té aquesta procedència i facilitarà la identificació de les que en podrien resultar afectades; al marge dels canvis en les subordinades **EXPL-SUBCONJ** i **EXPLORA-9+9+5**, l'assignació de valor a **XY** és l'única novetat que ens trobarem en **ACTUALITZA-PLUS**. Amb aquestes quatre pistes el lector ja en tindrà prou per interpretar el codi del bloc constituït per **ACTUALITZA**, **ACTUALITZA-PLUS**, **EXPLORA-9+9+5** i **EXPL-SUBCONJ**, i potser només en l'última funció convé fer una precisió a propòsit de l'expressió

```
(if (= (length L) K)
    (progn
      (setq P-DEP () OK ())
      (foreach E F (setq P-DEP (cons (nth E F-9) P-DEP)))
      (foreach M L
        (if (not OK)
            (mapcar '(lambda (E-9 E-L)
                       (if (and (not OK)
                                (not (member E-9 P-DEP))
                                (member M E-L))
                           (setq OK T L-DEP L PLUS T)))
                    F-9 F-L)))))
```

que sembla innecessàriament complicada perquè en aparença n'hauria hagut prou amb

```
(if (= (length L) K)
    (progn
      (setq L-DEP L PLUS T P-DEP ())
      (foreach E F
        (setq P-DEP (cons (nth E F-9) P-DEP)))))
```

Allò que sembla una complicació gratuïta no és més que una manera no excessivament onerosa (en termes de temps) d'evitar que cada cop que passem per **ACTUALITZA-PLUS** la detecció d'aquestes llistes dugui de nou a **ACT-DEPURA**, no només perquè la seva acció seria estèril sinó perquè entraríem en un bucle sense fi: cal deixar palès, en les inevitables "deteccions" posteriors, que la unitat ja havia estat depurada. A més, aquesta recapitulació de llistes tancades (de valors admissibles), l'entorn de les quals ja s'hi ha armonitzat, ens servirà per posar sota control uns efectes inesperats que, sent l'últim escull que quedava per salvar, abordarem al final del present subcapítol.

El codi que us oferim en les següents 4 pàgines i escaig correspon al nivell més primari d'implementació del que acabem d'exposar, en el sentit d'adaptar-ho a la estructura existent amb el mínim de modificacions. Així, les depuracions dutes a terme virtualment per **ACTUALITZA** (**REAL** = **nil**) són guardades a **TN/R-L** (encara que **TERCETS** no hagi introduït canvis a **R-LLL**) i recuperades per **ACTUALITZA** (**REAL** = **T**) en la primera part d'aquesta funció, quan el valor escollit és assignat realment. L'emmagatzament es fa des de **TERCETS**, seguint el conducte habitual per als canvis en **R-LLL** realitzats per aquesta última funció: transferint la informació a **TN/R-L** (parlem de la primera etapa funcional en la vida d'aquesta variable, que després aprofitem per a una nova missió) i tornant-la a recompondre mitjançant **RE-ACT-LLL**.

```
(defun INI-LLISTES (/ LA LL)
  (foreach N '(3 4 5 6 7 8 9)
    (setq LA ())
    (repeat N (setq N (1- N) LA (cons (list N) LA)))
    (setq LL LA)
    (repeat (1- (length LL))
      (setq LA LL LL ())
      (repeat (1- (length LA))
        (setq I (car LA) LA (cdr LA) J (reverse (cdr (reverse I))))
        (foreach E LA
          (if (equal (reverse (cdr (reverse E))) J)
              (setq LL (cons (append I (list (last E))) LL)))))
      (setq LL (reverse LL) LLL (cons LL LLL)))
    (setq LLL (reverse (cdr LLL)))
    (repeat (- (length (last LLL)) 3) (setq LLL (cdr LLL)))
    (setq LLISTES (cons LLL LLISTES)))
  (setq LLISTES (cdr (reverse LLISTES))))

(defun DEPURA (/ E)
  (foreach N (reverse E-L) (setq E (cons (if (member N L-DEP) () N) E))))

(defun ACT-DEPURA (/ XX YY)
  (if (not XY)
      (if (not (equal (setq XX (caar P-DEP)) (caadr P-DEP)))
          (setq XX () YY (cadar P-DEP))))
  (mapcar '(lambda (F-9 F-L)
             (mapcar '(lambda (E-9 E-L)
                        (if E-L
                            (if (member E-9 P-DEP)
                                E-L
                                (if (and XY (equal E-9 XY 1.1))
                                    (DEPURA)
                                    (if XX
                                        (if (= (car E-9) XX)
                                            (DEPURA)
                                            E-L)
                                        (if (and YY (= (cadr E-9) YY))
                                            (DEPURA)
                                            E-L))))))
                      F-9 F-L))
          A-9*9 A-LLL))

(defun TERCETS (<R-9 <R-L / R-9 R-L TRANSP TERC SEGUIR I J K K1 K2 TN)
  (foreach 3N 3*N
    (if (and (not TERC) (member N 3N))
        (progn
          (setq R-9 <R-9 R-L <R-L
                N1 (car 3N) N2 (cadr 3N) N3 (last 3N)
                TRANSP ())
          (repeat 2
            (if (not TERC)
                (progn
                  (setq I () J -1 K 0)
                  (mapcar '(lambda (F/R-9 F/R-L)
                             (setq J (1+ J) K1 0 K2 0)
                             (mapcar '(lambda (E/R-9 E/R-L)
                                        (if (and (or E/R-L (= N1 E/R-9)
                                                     (= N2 E/R-9) (= N3 E/R-9))
                                                 (or (not E/R-L) (member N1 E/R-L)
                                                     (member N2 E/R-L)
                                                     (member N3 E/R-L)))
                                            (setq K1 (1+ K1))
                                            (setq K2 (1+ K2))))
                                      F/R-9 F/R-L)
                             (if (= K1 3) (setq I J))
                             (if (= K2 3) (setq K (1+ K))))
                          R-9 R-L)
```

```
                    (if (and I (= K 2))
                        (progn
                          (foreach L (setq J (nth I R-L))
                            (setq K1 ())
                            (foreach E (setq K L)
                              (if (and E (not (member E 3N)))
                                  (setq TERC T K1 T K (subst () E K))))
                            (if K1 (setq J (subst K L J))))
                          (if TERC (setq R-L (subst J (nth I R-L) R-L)))
                          (setq J -1 K1 () SEGUIR T)
                          (foreach N (reverse L1A9)
                            (if (not (member N 3N)) (setq K1 (cons N K1))))
                          (while (and SEGUIR (< J 3))
                            (setq K2 K1 J (1+ J))
                            (if (/= J I)
                                (progn
                                  (mapcar '(lambda (E/R-9 E/R-L)
                                             (if E/R-L
                                                 (foreach N E/R-L
                                                   (if (member N K2)
                                                       (setq K2 (K2- N))))
                                                 (setq K2 (K2- E/R-9))))
                                          (nth J R-9) (nth J R-L))
                                  (if (= (length K2) 3)
                                      (progn
                                        (setq SEGUIR ())
                                        (foreach L (setq K (nth (- 3 (+ I J)) R-L)
                                                         K1 K)
                                          (if L
                                              (progn
                                                (setq I () J I)
                                                (foreach E L
                                                  (setq J
                                                        (cons
                                                          (if E
                                                              (if (member E K2)
                                                                  E
                                                                  (not (setq I T))))
                                                          J)))
                                                (if I (setq K1 (subst (reverse J)
                                                                      L K1))))))
                                        (if (not (equal K1 K))
                                            (setq R-L (subst K1 K R-L) TERC T)))))))
                            (if TERC (setq R-L (if TRANSP (TRANSP-3*3 R-L) R-L)
                                           TN/R-L (3*3/R-L TN/R-L) TN T REPAS T))))
                    (if (not (or TRANSP TERC))
                        (setq R-9 (TRANSP-3*3 R-9)
                              R-L (TRANSP-3*3 R-L) TRANSP T))))))))
  (if (and DEP (not TN)) (setq R-L <R-L TN/R-L (3*3/R-L TN/R-L))))

(defun PRESENTS ()
  (foreach E F
    (foreach M (nth E F-L)
      (if (not (member M L)) (setq L (cons M L))))))

 (defun EXPL-SUBCONJ (/ L OK)
  (setq K 1)
  (foreach Ñ (nth (- (length F-L) 4) LLISTES)
    (if (not (or PLUS FORA))
        (progn
          (setq K (1+ K))
          (foreach F Ñ
            (if (not (or PLUS FORA))
                (progn
                  (setq L ())
                  (if (< (1+ K) (length F-L))
                      (progn
                        (PRESENTS)
```

```
                        (if (= (length L) K)
                            (progn
                              (setq P-DEP () OK ())
                              (foreach E F (setq P-DEP (cons (nth E F-9) P-DEP)))
                              (foreach M L
                                (if (not OK)
                                    (mapcar '(lambda (E-9 E-L)
                                               (if (and (not OK)
                                                        (not (member E-9 P-DEP))
                                                        (member M E-L))
                                                   (setq OK T L-DEP L PLUS T)))
                                            F-9 F-L)))))))
                    (if (and (not FORA) (> K 2))
                        (progn
                          (if (not L) (PRESENTS))
                          (if (< (length L) K) (setq FORA T)))))))))))

(defun EXPLORA-9+9+5 (/ NN MN F-9 F-L)
  (foreach Ñ L1A9 (if (not (member Ñ F-9*9)) (setq NN (cons Ñ NN))))
  (if (> (length NN) 0)
      (progn
        (setq NN (reverse NN) MN (last NN))
        (foreach Ñ NN
          (if (not PLUS)
              (progn
                (setq K 0)
                (if (and INI (= Ñ MN)) (setq F-9 () F-L ()))
                (mapcar '(lambda (E-9*9 E-LL3)
                           (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
                           (if (and INI E-LL3 (= Ñ MN))
                               (progn
                                 (setq I ())
                                 (foreach M E-LL3 (if M (setq I (cons M I))))
                                 (setq F-9 (cons E-9*9 F-9) F-L (cons I F-L)))))
                        F-9*9 F-LL3)
                (if (= K 1)
                    (PLUS-N->P J)
                    (if (= K 0)
                        (setq FORA T)
                        (if (and INI (= Ñ MN) (> (length F-L) 3))
                            (EXPL-SUBCONJ)))))))))))

(defun ACTUALITZA-PLUS (/ I L M Ñ X1 Y1 5P R-9 R-L ORIG)
  (setq FORA ())
  (mapcar '(lambda (9*9 LL3)
             (setq ORIG (not ORIG))
             (if (not (or PLUS FORA))
                 (mapcar '(lambda (F-9*9 F-LL3)
                            (if (and (not (or PLUS FORA)) ORIG)
                                (mapcar '(lambda (E-9*9 E-LLL)
                                           (if (and (not (or PLUS FORA)) E-LLL)
                                               (progn
                                                 (setq K 0)
                                                 (foreach E E-LLL
                                                   (if (and (< K 2) E)
                                                       (setq Ñ E K (1+ K))))
                                                 (if (= K 1)
                                                     (PLUS-N->P E-9*9)
                                                     (if (= K 0)
                                                         (setq FORA T))))))
                                        F-9*9 F-LL3))
                            (if (not (or PLUS FORA)) (EXPLORA-9+9+5)))
                          9*9 LL3)))
           (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL)))
  (if (not (or PLUS FORA))
      (progn
        (setq X1 (* (/ X 3) 3) Y1 (* (/ Y 3) 3) ORIG ())
        (foreach K '(6 3 0) (if (/= K Y1) (setq 5P (cons (list X1 K) 5P))))
```

```
        (foreach K '(6 3 0) (setq 5P (cons (list K Y1) 5P)))
        (repeat (if INI 2 1)
          (setq ORIG (not ORIG))
          (if (not (or PLUS FORA))
              (foreach Y '(0 3 6)
                (foreach X '(0 3 6)
                  (if (and ORIG (not (or PLUS FORA)) (member (list X Y) 5P))
                      (progn
                        (setq XY (list (1+ X) (1+ Y))
                              R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              5P (subst (cons (list X Y) (list R-9 R-L))
                                                         (list X Y) 5P))
                        (mapcar '(lambda (F-9*9 F-LL3)
                                   (if (not (or PLUS FORA))
                                       (EXPLORA-9+9+5)))
                                (list (append (car R-9) (cadr R-9) (last R-9)))
                                (list (append (car R-L) (cadr R-L) (last R-L))))))
                  (if (not ORIG)
                      (progn
                        (if (setq R-L (assoc (list X Y) 5P))
                            (setq R-9 (cadr R-L) R-L (caddr R-L))
                            (setq R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))
                                  R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))))
                        (setq K 0)
                        (foreach F R-9
                          (foreach E F (if (atom E) (setq K (1+ K)))))
                        (if (< K 9) (TERCETS R-9 R-L))
                        (setq N/R-L (3*3/R-L N/R-L)))))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y XY L PLUS P-DEP L-DEP DEP)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM" "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P))
             (if L-DEP
                 (setq A-LLL (ACT-DEPURA) DEP T L-DEP () XY ())
                 (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
                       A-LLL (ACT-LLL X Y A-LLL) XY ()))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA REPAS 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq TT (cdr (assoc N TN/R-L))
                    R-9*9 (car 2R) R-LLL (if TT (RE-ACT-LLL) (cadr 2R))
                    R-LLL (if REPAS (FILS/COLS) R-LLL))
              (if TT (setq TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
```

```
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (and (not OK) (member E R-N))
              (progn
                (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                (if (not FORA)
                    (if (equal (cadr 2R) NIL*NIL)
                        (setq OK T)
                        (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

És clar que, fet això, un es pregunta per què, si no apareix cap configuració **AAA-AAA** o **AAA-BBB**, ens hem de molestar a guardar els canvis a **R-LLL** deguts a **DEPURA** < < **ACT-DEPURA**. ¿No seria menys complicat desampallegar-se d'aquesta informació una vegada ens ha permés arribar sense entrebancs a sengles veredictes d'idoneïtat per a cadascun dels valors candidats a la casella actual, igual com fem amb els forats tapats en el transcurs de les exploracions (i en les fases **INI**cials, quan el pas per **TERCETS** no ha tingut conseqüències), acceptant reconstruir-la del no-res si el valor finalment assignat defineix algun conjunt de caselles tancat als efectes de valors admissibles? Així, les depuracions realitzades virtualment per **ACTUALITZA** (**REAL** = **nil**) no es guardarien a **TN/R-L** (tret que **TERCETS** hagués comportat canvis a **R-LLL**) sinó que es perdrien, i **ACTUALITZA** (**REAL** = **T**) partiria de zero en assignar realment un valor. Per tal que aquesta funció pogués fer aquesta tasca en la seva segona part (**REAL** = **T** i **TT** = **nil**), caldria ampliar els accessos a **EXPL-SUBCONJ** des d'**EXPLORA-9+9+5** < **ACTUALITZA-PLUS**, substituint la condició **(and INI (= Ñ MN))** per **(and (or INI REAL) (= Ñ MN))**, mentre que **ACTUALITZA** únicament es diferenciaria per haver prescindit de la variable **DEP** i **RE-MEMBER** de la variable **REPAS**. La resta de codi queda igual (**TERCETS** retorna a la penúltima versió i tampoc la representem):

```
(defun EXPLORA-9+9+5 (/ NN MN F-9 F-L)
  (foreach Ñ L1A9 (if (not (member Ñ F-9*9)) (setq NN (cons Ñ NN))))
  (if (> (length NN) 0)
      (progn
        (setq NN (reverse NN) MN (last NN))
        (foreach Ñ NN
          (if (not PLUS)
              (progn
                (setq K 0)
                (if (and (or INI REAL) (= Ñ MN)) (setq F-9 () F-L ()))
                (mapcar '(lambda (E-9*9 E-LL3)
                           (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
                           (if (and (or INI REAL) E-LL3 (= Ñ MN))
                               (progn
                                 (setq I ())
                                 (foreach M E-LL3 (if M (setq I (cons M I))))
                                 (setq F-9 (cons E-9*9 F-9) F-L (cons I F-L)))))
                        F-9*9 F-LL3)
                (if (= K 1)
                    (PLUS-N->P J)
                    (if (= K 0)
                        (setq FORA T)
                        (if (and (or INI REAL) (= Ñ MN) (> (length F-L) 3))
                            (EXPL-SUBCONJ)))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y XY L PLUS P-DEP L-DEP)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM" "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
```

```
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P))
             (if L-DEP
                 (setq A-LLL (ACT-DEPURA) L-DEP () XY ())
                 (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
                       A-LLL (ACT-LLL X Y A-LLL) XY ()))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R) TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (subst (list N (REORD R-LLL) (REORD REPLUS))
                                         (cons N TT) TN/R-L)))))))
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (and (not OK) (member E R-N))
              (progn
                (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                (if (not FORA)
                    (if (equal (cadr 2R) NIL*NIL)
                        (setq OK T)
                        (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

Tot i que, ben pensat, encara hi ha una solució més neta (no parlem de eficiència perquè els temps són pràcticaments idèntics): guardar a **TN/R-L** les depuracions realitzades virtualment per **ACTUALITZA** (**REAL** = **nil**), encara que **TERCETS** no hagi introduït cap modificació a **R-LLL**, i recuperar-les en la primera part d'aquesta funció, quan s'assigna realment el valor escollit (**REAL** = **T**); l'emmagatzament no es produeix des de **TERCETS** sinó a **RE-MEMBER**, assignant **R-LLL** directament a **TN/R-L** en la segona etapa funcional d'aquesta variable, i així ens estalviem la feina de descompondre-la en requadres 3×3 (amb **SUB-X*Y**, format obligat per a la detecció de configuracions **AAA-AAA** i **AAA-BBB** però no per a la cerca de conjunts tancats de valors admissibles, localitzats per files o columnes la major part de vegades), transferir-la a **TN/R-L** quan viu la seva primera etapa i tornar-la a recompondre amb **RE-ACT-LLL**. Limitant-nos a presentar els blocs de codi que hagin experimentat algun canvi (**EXPLORA-9+9+5** retorna a la penúltima versió i tampoc la representem), encara que siguin tan nimis com la reaparició de la variable **DEP**, que ha passat de ser local en **ACTUALITZA** a ser-ho a **RE-MEMBER**), tot plegat quedarà així:

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y XY L PLUS P-DEP L-DEP)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM" "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P))
             (if L-DEP
                 (setq A-LLL (ACT-DEPURA) DEP T L-DEP () XY ())
                 (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
                       A-LLL (ACT-LLL X Y A-LLL) XY ()))
```

```
            (ACTUALITZA-PLUS)
            (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R) TT (cdr (assoc N TN/R-L)))
              (if (or TT DEP)
                  (progn
                    (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)))
                    (setq 2R (list N (REORD R-LLL) (REORD REPLUS))
                          TN/R-L (if TT
                                     (subst 2R (cons N TT) TN/R-L)
                                     (if TN/R-L
                                         (cons 2R TN/R-L)
                                         (list 2R))))))))))
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (and (not OK) (member E R-N))
              (progn
                (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                (if (not FORA)
                    (if (equal (cadr 2R) NIL*NIL)
                        (setq OK T)
                        (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

Aquesta versió serà la de referència a totes les esmenes que d'ara endavant haguem d'introduir. Què vol dir això? -es preguntarà el lector, a qui justificadament li haurà pujat la mosca al nas- ¿Que potser encara no som al cap del carrer, almenys pel que fa a l'opció A de **ABC** ("SOLUCIÓ CREADA")? Doncs no, dissortadament: com ja havia passat altres vegades, en verificar el funcionament de la nova versió sobre l'última tanda d'exemples, més enllà dels casos amb què obríem aquest subcapítol, mantenint el bon costum de no deturar la verificació un cop superat el punt crític i prosseguint l'emplenament de l'escaquer 9×9 fins a completar la SUDOKU-SOLUCIÓ, han tornat a produir-se esperes que ultrapassaven allò que considerem acceptable. A continuació us presentem tres casos apareguts en seguir jugant amb un d'aquells exemples: com de costum, els subratllats caracteritzen la casella actual amb un valor candidat que està trigant més del compte a rebre el veredicte (que en tots tres casos és d'aptitud), i a més hem marcat amb el signe "@" les caselles lliures que formen part de subconjunts tancats de candidats (subconjunts que normalment completen línies -passa en els tres-, columnes i/o requadres 3×3 -només passa en el primer-), per tal de copsar més ràpidament la seva disposició a l'escaquer 9×9.

```
0 0 0   7 5 0   8 @ 0       0 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
0 0 5   0 0 0   0 7 0       0 0 5   0 0 0   7 1 0       0 0 5   0 0 0   7 1 0
0 0 4   0 0 0   5 @ 0       0 0 4   0 0 0   5 0 0       0 0 4   0 0 0   5 0 0

@ 5 @   @ @ @   4 @ @       @ @ @   @ @ @   4 @ @       @ @ @   @ @ @   4 @ @
@ @ @   4 @ 5   @ 8 @       0 0 0   4 0 5   0 0 0       0 0 0   4 0 5   0 0 0
@ @ @   1 2 3   @ @ @       0 0 0   1 2 3   0 0 0       0 0 0   1 2 3   0 0 0

0 0 3   0 0 0   1 @ 0       0 0 3   0 0 0   1 0 0       0 0 3   0 0 0   1 0 0
8 0 2   0 0 0   3 @ 0       0 0 2   0 0 0   3 0 0       0 0 2   0 0 0   3 0 0
0 0 1   0 0 0   2 @ 8       0 0 1   0 0 0   2 0 0       0 0 1   0 0 0   2 0 0
```

Abans d'entrar en l'estudi del nou entrebanc, convé descriure aquests subconjunts (són disjunts en l'àmbit de cada unitat modular, però tenen caselles compartides), tot recordant que els **n** valors definitoris no sempre apareixen a totes **n** caselles. I de nou ens hem de disculpar per una compaginació que dificultarà el seguiment de les descripcions (en el full següent) sobre aquestes tres representacions.

En el primer cas (esquerra), identificarem:
- A la 5ª fila: el conjunt compost per les caselles 1,5, 2,5 i 9,5, i definit pels valors potencials **1**, **2** i **3**; el conjunt compost per les caselles 3,5, 5,5 i 7,5, i definit pels valors potencials **6**, **7** i **9**.
- A la 6ª fila: el conjunt compost per les caselles 1,6, 8,6 i 9,6, i definit pels valors potencials **1**, **2** i **3**; el conjunt integrat per les caselles 3,6, 4,6, 5,6 i 6,6, i definit pels valors potencials **6**, **7**, **8** i **9**.
- A la 7ª columna: el conjunt compost per les caselles 7,1, 7,2, 7,3 i 7,4, i definit pels valors potencials **4**, **5**, **6** i **9**; el conjunt compost per les caselles 7,6, 7,7 i 7,9, i definit pels valors potencials **1**, **2** i **3**.
- En el requadre 3×3 centre-esquerra: el conjunt compost per les caselles 1,4, 2,4, 3,4, 3,5 i 3,6, i definit pels valors potencials **4**, **6**, **7**, **8** i **9**; el conjunt compost per caselles 1,5, 2,5 i 1,6, i definit pels valors potencials **1**, **2** i **3**.
- En el requadre 3×3 central: el conjunt compost per les caselles 5,5, 4,6, 5,6 i 6,6, i definit pels valors potencials **6**, **7**, **8** i **9**.
- En el requadre 3×3 centre-dreta: el conjunt compost per les caselles 7,4, 8,4, 9,4 i 7,5, i definit pels valors potencials **5**, **6**, **7** i **9**; el conjunt compost per les caselles 9,5, 8,6 i 9,6, i definit pels valors potencials **1**, **2** i **3**.

En el segon i tercer cas (centre i dreta), idèntics pel que fa a aquests conjunts:
- A la 6ª fila: el conjunt compost per les caselles 1,6, 2,6, 8,6 i 9,6, i definit pels valors potencials **1**, **2**, **3** i **5**; el conjunt integrat per les caselles 3,6, 4,6, 5,6 i 6,6, i definit pels valors potencials **6**, **7**, **8** i **9**.

Amb el desconcert inicial, la primera reacció de l'autor va ser decidir que potser era el moment d'invertir el mètode heurístic: si fins ara havíem progressat fent front a les incidències que s'anaven presentant i sovint ens adonàvem *a posteriori* que els dispositius implementats responien a recursos utilitzats en la resolució manual dels sudokus, ara que ja començàvem a estar una mica farts de navegar en la direcció teòricament correcta però sense albirar encara el final de la travessia, ¿no seria profitós fer inventari d'aquests recursos i mirar si en queda algun que no hagi estat incorporat? Per exemple, podríem arrodonir l'acció d'**ACTUALITZA-PLUS** amb la detecció d'una configuració no massa freqüent però que cap jugador habitual desaprofitaria sense emplenar una casella: quan la presència d'un valor admissible exclusivament en 2 caselles d'una mateixa fila [columna] i requadre 3×3 també es dóna en una altra fila [columna] de manera que les 4 caselles estan situades en la mateixa columna [fila] i, per aquesta raó, les 2 últimes també pertanyen al mateix requadre 3×3; en aquestes circumstàncies, si en el requadre alineat amb els altres dos també hi ha caselles amb aquest valor admissible que pertanyen exclusivament a la mateixa fila [columna], a la que no se situa sobre cap de les columnes [files] esmentades hi ha de figurar necessàriament aquest valor. Ara bé, abans d'embarcar-nos en la preparació d'un dispositiu que difícilment es podrà improvisar a partir del codi acual sinó que més aviat durà feina, convindria tenir presents dos fets: que, com veurem de seguida, hem arribat a un punt on totes les optimitzacions que no apuntin directament a evitar que alguns veredictes encara es facin el ronso, és millor que les deixem córrer, perquè a la pràctica només servirien per alentir la resposta en totes les assignacions; en cap dels tres casos esmentats es presenta aquesta configuració (no ens referim a emplenaments posteriors sinó a la situació d'*impasse* que ens ocupa), així que de ben poc serviria incorporar al programa un dispositiu com aquest de què parlàvem. Així que no hi ha més remei que seguir fent com sempre: esbrinar l'origen de les demores actuals i tractar de posar-hi remei.

Abans de res, vejam quines són aquestes demores inacceptables: en el primer cas la casella que anem a emplenar és 8,5, l'únic candidat que fa el ronso és el que hi figura (**8**) i el veredicte (favorable) triga 3 minuts i mig, mentre que els altres veredictes són pràcticament instantanis; en el segon, la casella actual és 8,8 i els veredictes reticents (**1**, **2** i **3**) triguen un minut cadascun; en el tercer, la casella actual és 7,8 i els veredictes reticents (**7** i **8**) també triguen un minut. Tots tres casos, però, tenen una cosa en comú que ens situarà sobre la pista de l'origen del mal: l'activació de les caselles subratllades suposa que el requadre de 27 caselles més poblat ja no serà un requadre 9×3 (amb 8 caselles ocupades en el primer, i amb 6 en el segon i tercer) sinó un 3×9 (amb 9 caselles ocupades en el primer, i amb 7 en el segon i tercer), cosa que comportarà la transposició de l'escaquer (i, si s'escau, permutacions de requadres 9×3 i de files dins de cada un d'aquests requadres), teòricament per agilitar les exploracions a què **RE-MEMBER** sotmet els candidats a aquestes caselles. Heus-los aquí a punt per a l'exploració:

```
0 8 0   @ @ @   0 0 0       0 0 0   0 0 @   0 0 0       0 0 0   0 0 @   0 0 0
0 0 0   @ @ 5   0 0 0       0 0 0   0 0 @   0 0 0       0 0 0   0 0 @   0 0 0
1 2 3   @ @ @   4 5 0       1 2 3   0 0 @   4 5 0       1 2 3   0 0 @   4 5 0

0 0 0   3 5 @   0 0 0       0 0 0   2 0 @   0 0 0       0 0 0   2 0 @   0 0 0
0 0 0   2 @ @   0 0 5       0 0 0   3 5 @   0 0 0       0 0 0   3 5 @   0 0 0
0 0 0   1 4 @   0 0 7       0 0 0   1 4 @   0 0 0       0 0 0   1 4 @   0 0 0

0 0 0   @ @ @   0 0 0       0 0 0   0 0 @   0 0 0       0 0 0   0 0 @   0 0 0
@ @ @   @ 8 @   @ 7 @       0 0 0   0 0 @   0 1 0       0 0 0   0 0 @   0 1 0
2 3 1   @ @ 4   5 0 8       2 3 1   0 0 4   5 7 0       2 3 1   0 0 4   5 7 0
```

Paradoxalment, si intervenim la primera part de **MASCARA** per tal d'evitar aquesta transposició (i, de passada, les permutacions de requadres 9×3 i les de files dins de cada un d'aquests requadres, de forma que **RE-MEMBER** < **MASCARA** treballi amb les pseudomatrius ****9*9**** i **LLL** originals), tots els veredictes arribaran puntualment. Si recuperem el dispositiu de recompte de retrocessos, agrupats per requadres 9×3, que havíem posat en marxa a *Etapa prèvia: crear una solució, III*, la primera cosa a destacar és que, en general i per a una mateixa durada, el nombre de retrocessos ha baixat, si ho compareu amb aquells primers resultats. La segona és que, com era previsible i trobareu confirmat en els tres parells de llistats que ara us oferim, els temps de processat es corresponen al nombre de retrocessos, no sols pel que fa a les diferències entre candidats (malgrat ser tots admissibles) sinó avaluant-los en dues situacions: amb **RE-MEMBER** treballant sobre un escaquer en què requadres i files s'han permutat, transposat (a l'esquerra) i sense transposició (a la dreta). No cal dir que la coincidència entre els retrocessos corresponents al candidat 1> del segon parell de llistats i els corresponents al candidat 7> del tercer parell no té res de casual: es tracta d'exploracions idèntiques, on és irrellevant que en un cas l'emplenament de la casella 8,1 [la 7,8] sigui real i el de la 8,2 [la 8,8] virtual, i en l'altre passi exactament a l'inrevés.

```
1>     0 +     1 + 0 =     1          1> 0 +  0 + 0 =  0
2>     0 +     1 + 0 =     1          2> 0 +  0 + 0 =  0
3>     0 +     0 + 0 =     0          3> 0 +  0 + 0 =  0
6>     0 +     0 + 0 =     0          6> 0 +  0 + 0 =  0
8> 2.288 + 9.366 + 0 = 11.654          8> 0 +  0 + 0 =  0
9>     0 +     0 + 0 =     0          9> 0 +  0 + 0 =  0
T> 2.288 + 9.368 + 0 = 11.656          T> 0 +  0 + 0 =  0

1>    94 + 4.440 + 0 =  4.534          1> 1 +  6 + 0 =  7
2>    94 + 4.280 + 0 =  4.374          2> 1 +  6 + 0 =  7
3>    94 + 4.212 + 0 =  4.306          3> 1 +  6 + 0 =  7
4>     0 +     0 + 0 =      0          4> 0 + 12 + 0 = 12
6>     0 +     0 + 0 =      0          6> 1 +  0 + 0 =  1
8>     0 +     0 + 0 =      0          8> 1 +  2 + 0 =  3
9>     0 +     0 + 0 =      0          9> 1 + 12 + 0 = 13
T>   282 + 12.932 + 0 = 13.214          T> 6 + 44 + 0 = 50

6>     0 +     0 + 0 =      0          6> 1 +  3 + 0 =  4
7>    94 + 4.440 + 0 =  4.534          7> 1 +  6 + 0 =  7
8>    94 + 4.440 + 0 =  4.534          8> 1 +  2 + 0 =  3
9>     0 +     0 + 0 =      0          9> 1 +  6 + 0 =  7
T>   188 + 8.880 + 0 =  9.068          T> 4 + 17 + 0 = 21
```

La davallada de la ràtio durada/retrocessos en les exploracions **RE-MEMBER**, que ha estat realment espectacular, com dèiem línies amunt, no ens ha de sorprendre pas: tingueu present que, en els primers recomptes, els accessos a **ACTUALITZA** amb **INI** = = **nil** (és a dir, els de totes les autorecursions de **RE-MEMBER**) no contemplaven la intervenció d'**ACTUALITZA-PLUS**; més tard, aquests accessos ja la incloïen, però amb l'acció tapaforats contextual limitada als requadres 3×3; no ha estat fins fa poc, entre les versions que se succeïen en el subcapítol precedent, que vam prendre la decisió d'ampliar el rastreig de forats a files i columnes. I ara, per una banda tenim unes recursions més entretingudes (fa poc dèiem que havíem arribat a un punt d'inflexió en què no ens podíem permetre més "optimitzacions"), però per l'altra, a mesura que s'emplena virtualment l'escaquer, augmenta el nombre d'assignacions automàtiques per tapar forats, amb la qual cosa disminueix el nombre de caselles que ens cal transitar, el de valors admissibles en primera instància a cadascuna d'elles i, per tant, la probabilitat de retrocessos i de veredictes desfavorables.

Ara, un cop feta aquesta precisió, seria convenient rebobinar la película (de la qual us hem desvetllat pràcticament el desenllaç) i passar-la de nou però a càmara lenta, il·lustrant-la amb les reflexions de l'autor a cada descobriment, perquè estem davant d'un cas curiós en què identificar empíricament les circumstàncies en què es produeixen les llargues esperes i trobar-ne el remei ha estat relativament senzill, però no ho ha estat tant aclarir la relació lògica entre causa i efecte.

Com que us pot estranyar que els tres exemples siguin tan semblants, convé aclarir que va ser el presentat en primer lloc aquell on, en la tasca de consolidació de les últimes modificacions introduïdes (ja s'ha dit que anàvem comprovant si també rutllaven sobre exemples anteriors als que n'havien donat origen, no deturant-nos en les respectives situacions crítiques sinó aprofitant l'avinentesa per completar l'emplenament de l'escaquer), el procés va encallar inesperadament. Els altres dos els vam anar a buscar: el desconcert inicial (i la temptació d'incorporar recursos diversos, habitualment explotats en la resolució manual de sudokus, fins que algun ens resolgués el problema, temptació sortosament reprimida) va fer que seguíssim donant voltes sobre el mateix exemple, descobrint que n'hi havia prou a introduir els valors 1, 2, 3, 4 i 5 a la 3ª i 7ª columna i en el requadre 3×3 central perquè prenguessin cos els dos subconjunts tancats de candidats que completen la 6ª fila (parlem de la versió original, tres pàgines enrera); va ser així com, ocupant una casella més i activant-ne un altre en el requadre 3×3 superior dret, vam descobrir que en aquests dos casos, més senzills que el primer, el pronunciament sobre certs candidats també s'eternitzava. Per contra, en altres de molt semblants no passava res: si la casella 8,8 l'emplenàvem amb un 7, com en el primer cas, i activàvem la casella 7,8, els veredictes (també favorables) sobre els candidats 6, 8 i 9 eren gairebé instantanis (aquesta variant és fàcil d'imaginar i no l'hem representada).

Va ser en aquests casos derivats del primer i amb ocupacions idèntiques (únicament difereixen en l'ordre de les dues últimes, i això sols repercuteix en la identitat dels candidats) on de sobte va venir la inspiració: l'activació (ocupació virtual) de l'última casella originava la transposició de les pseudomatrius 9×9 de treball, en passar el requadre 3×9 dret a estar més poblat que el més poblat dels requadres 9×3; i això mateix passava en el primer cas, el que havia disparat l'alarma, quan activàvem 8,5. Vist això, no calia ser cap Einstein per fer la prova definitiva: inhibint el dispositiu causant de la transposició, teòricament optimitzador de les exploracions **RE-MEMBER**, es va comprovar com tots els veredictes ja es produïen en un temps raonable. Tanmateix, identificats els símptomes de la malaltia i fins i tot l'etiologia, i verificada en tots els casos l'eficàcia del remei, quedaba un petit detall: saber quin mecanisme desencadenava la primera (esperes anormalment llargues en alguns candidats) a partir de la segona (l'existència de subconjunts tancats de valors admissibles, organitzats en columna). Si ja en el primer exemple s'ens hagués acudit que l'activació de la casella 8,5 comportava la transposició automàtica de les matrius 9×9, potser hauríem perdut encara més temps en atribuir directament el nou contratemps a l'existència de subconjunts tancats de candidats abastant (juntament amb les caselles ocupades) un requadre 9×3 sencer (que en la transposició es transformava en requadre 3×9), perquè d'alineacions simples també n'hi havia (la 8ª columna, que es convertia en 2ª fila en transposar-se l'escaquer i experimentar a més a més permutacions de files) però ja n'hi havia hagut abans: precisament la novetat aportada per 8,5 (**8**) era que el subconjunt preexistent que coincidia amb la 6ª fila s'hagués engreixat fins a cobrir el requadre 9×3 del mig. Sortosament, la descoberta dels altres dos casos ens va lliurar per una vegada de perdre el temps amb pressumpcions falses: quedava clar que n'hi havia prou amb un conjunt-fila que passés a conjunt-columna per causar problemes, i en el primer cas podia interpretar-se que l'efecte de tres conjunts-fila passant a conjunts-columna predominava sobre el d'un conjunt-columna que passés a conjunt-fila.

Quant al mecanisme, s'intuïa que la depuració practicada en els elements de **R-LLL**, reduint a l'indispensable el nombre de valors potencials de les caselles buides (en definitiva, eliminant la redundància d'aquest bloc d'informació), ja suposa un avenç considerada individualment, casella a casella, però que en constituir amb les caselles afectades subconjunts excloents donem un pas endavant qualitativament important, perquè l'estalvi (de recorregut, en l'exploració **RE-MEMBER**) propiciat per aquests subconjunts és superior a la suma dels estalvis deguts a cada casella, anàlogament a la resistència d'una corda composta per fibres torçades o trenades, molt superior a la suma de les resistències de cada fibra. Però aquest capteniment com a sistema únicament es dóna quan els subconjunts tancats s'organitzen en fila, ja què és per files com s'estructura l'exploració de l'escaquer, i l'avaluació del

guany obtingut (decrement en el nombre permutacions de valor possibles) és fàcil. Ara bé: quan passem de considerar una fila aïllada a situar-la en el seu context, el problema de quantificar les trajectòries possibles en cada cas assoleix un grau de complexitat que ha superat l'autor (que amb prou feines se n'ha sortit adoptant unes simplificacions que invalidaven les conclusions, de tan grolleres com eren); no diguem ja de quantificar l'efecte d'una transposició que converteixi en columna una fila composta de subconjunts tancats de candidats... Així que l'autor, com de costum, ha deixat constància del seu fracàs en aquesta pàgina i les dues següents, tot convidant el lector a saltar-se-les.

Quan totes les caselles lliures d'una fila s'organitzen en subconjunts disjunts, a efectes de valors admissibles, i limitant-nos als casos de 2 subconjunts (un, de **m** caselles i **m** valors candidats, i l'altre, de **n** caselles i **n** valors candidats, sent $\mathbf{4 \le m + n \le 9}$), que són els més freqüents, el nombre de permutacions dels $\mathbf{m + n}$ candidats sobre les $\mathbf{m + n}$ caselles lliures serà no superior a $\mathbf{m!\,n!}$. Si, per anar a la hipòtesis més probable, la de màxima uniformitat en la ocupació de caselles, suposem que la resta de files i totes les columnes tenen igualment $\mathbf{m + n}$ caselles lliures, tot i no contenir subconjunts tancats, si transposem l'escaquer 9×9 i els dos subconjunts **m** i **n** passen situar-se en columna, el nombre de permutacions dels $\mathbf{m + n}$ candidats sobre les $\mathbf{m + n}$ caselles lliures, en qualsevol de les alineacions que abans eren columnes i ara són files, serà no superior a $\mathbf{(m + n)!}$ si la casella intersecció amb la columna esmentada (l'anterior fila amb els subconjunts **m** i **n**) està ocupada, a $\mathbf{m \times (m + n - 1)!}$ si la casella pertany al subconjunt de **m** candidats, o a $\mathbf{n \times (m + n - 1)!}$ si pertany al de **n** candidats. Si ara tenim en compte que $\mathbf{m > 1}$ i $\mathbf{n > 1}$, veureu com el nombre de possibilitats a explorar en una fila ordinària és superior al nombre de possibilitats en una fila amb el mateix nombre de caselles lliures però organitzades en dos subconjunts tancats de candidats (o si ho voleu en termes més afins als nostres interessos, com els temps d'exploració d'una fila ordinària són superiors als d'una fila amb el mateix nombre de caselles lliures però organitzades en dos subconjunts tancats de candidats), i això succeeix quan transposem un escaquer 9×9 amb una distribució uniforme de caselles ocupades i que té una fila d'aquestes característiques: la fila que apareix en la mateixa posició necessitarà més temps per ser explorada, tan si la casella que comparteix amb la que ha passat a ser columna està ocupada com si no (i en aquest últim cas, tant és que pertanyi al subconjunt **m** o **n**).

En efecte, $\mathbf{(m + n) \times (m + n - 1) \times \ldots \times (m + 1) > n!}$, perquè el primer membre no és sinó un factorial de **n** on hem sumat **m** a cada factor. I, si ara multipliquem per $\mathbf{m!}$ els dos membres de la desigualtat, tindrem $\mathbf{(m + n) \times (m + n - 1) \times \ldots \times (m + 1) \times m! > m!\,n!}$. Però $\mathbf{(m + n) \times (m + n - 1) \times \ldots \times (m + 1) \times m! = (m + n)!}$, o sigui que, quan la intersecció de la fila ordinària amb la columna que ha resultat de transposar la fila amb els subconjunts **m** i **n** sigui una casella ocupada, serà $\mathbf{(m + n)! > m!\,n!}$.

D'altra banda, $\mathbf{[(m - 1) + n] \times [(m - 1) + n - 1] \times \ldots \times [(m - 1) + 1] > n!}$, perquè el primer membre no és sinó un factorial de **n** on hem sumat $\mathbf{(m - 1)}$ a cada factor.
I, si ara multipliquem per $\mathbf{(m - 1)!}$ els dos membres de la desigualtat, tindrem que $\mathbf{[(m - 1) + n] \times [(m - 1) + n - 1] \times \ldots \times [(m - 1) + 1] \times (m - 1)! > (m - 1)! \times n!}$.
Però $\mathbf{[(m - 1) + n] \times [(m - 1) + n - 1] \times \ldots \times [(m - 1) + 1] \times (m - 1)! = [(m - 1) + n]!}$, o sigui que $\mathbf{[(m - 1) + n]! > (m - 1)! \times n!}$ Si ara multipliquem per **m** aquests dos membres, tindrem $\mathbf{m \times [(m - 1) + n]! > m \times (m - 1)! \times n! = m!\,n!}$, o sigui que, quan la intersecció de la fila ordinària amb la columna que ha resultat de transposar la fila amb els subconjunts **m** i **n** sigui una casella pertanyent al subconjunt de **m** candidats, també serà $\mathbf{m \times (m + n - 1)! > m!\,n!}$.

Anàlogament, quan la intersecció de la fila ordinària amb la columna resultant de transposar la fila amb els subconjunts **m** i **n** sigui una casella pertanyent al de **n** candidats, també serà $\mathbf{n \times (m + n - 1)! > m!\,n!}$

Atenint-nos a aquestes expressions, podríem quantificar les diferències per tenir noció del seu ordre de magnitud:
- En el cas que la fila tingui les 9 caselles lliures:
  - Conjunts tancats de 7 i 2 (caselles i valors potencials):
    - Suposant que la casella interceptada només té 7 valors potencials: 272.160
    - Suposant que la casella interceptada només té 2 valors potencials: 70.560
  - Conjunts tancats de 6 i 3 (caselles i valors potencials):

    - Suposant que la casella interceptada només té 6 valors potencials: 237.600
    - Suposant que la casella interceptada només té 3 valors potencials: 116.640
  - Conjunts tancats de 5 i 4 (caselles i valors potencials):
    - Suposant que la casella interceptada només té 5 valors potencials: 198.720
    - Suposant que la casella interceptada només té 4 valors potencials: 158.400
- En el cas que la fila tingui 8 caselles lliures:
  - Conjunts tancats de 6 i 2 (caselles i valors potencials):
    - Suposant que la casella interceptada és justament l'única ocupada: 38.880
    - Suposant que la casella interceptada només té 6 valors potencials: 28.800
    - Suposant que la casella interceptada només té 2 valors potencials: 8.640
  - Conjunts tancats de 5 i 3 (caselles i valors potencials):
    - Suposant que la casella interceptada és justament l'única ocupada: 39.600
    - Suposant que la casella interceptada només té 5 valors potencials: 24.480
    - Suposant que la casella interceptada només té 3 valors potencials: 14.400
  - Conjunts tancats de 4 i 4 (caselles i valors potencials):
    - Suposant que la casella interceptada és justament l'única ocupada: 39.744
    - Suposant que la casella interceptada té aquests valors potencials: 19.584
- En el cas que la fila tingui 7 caselles lliures:
  - Conjunts tancats de 5 i 2 (caselles i valors potencials):
    - Suposant que la casella interceptada és de les que estan ocupades: 4.800
    - Suposant que la casella interceptada només té 5 valors potencials: 3.360
    - Suposant que la casella interceptada només té 2 valors potencials: 1.200
  - Conjunts tancats de 4 i 3 (caselles i valors potencials):
    - Suposant que la casella interceptada és de les que estan ocupades: 4.896
    - Suposant que la casella interceptada només té 4 valors potencials: 2.736
    - Suposant que la casella interceptada només té 3 valors potencials: 2.016
- En el cas que la fila tingui 6 caselles lliures:
  - Conjunts tancats de 4 i 2 (caselles i valors potencials):
    - Suposant que la casella interceptada és de les que estan ocupades: 672
    - Suposant que la casella interceptada només té 5 valors potencials: 432
    - Suposant que la casella interceptada només té 2 valors potencials: 192
  - Conjunts tancats de 3 i 3 (caselles i valors potencials):
    - Suposant que la casella interceptada és de les que estan ocupades: 684
    - Suposant que la casella interceptada té aquests valors potencials: 324
- En el cas que la fila tingui 5 caselles lliures:
  - Conjunts tancats de 3 i 2 (caselles i valors potencials):
    - Suposant que la casella interceptada és de les que estan ocupades: 108
    - Suposant que la casella interceptada només té 3 valors potencials: 60
    - Suposant que la casella interceptada només té 2 valors potencials: 36
- En el cas que la fila tingui 4 caselles lliures:
  - Conjunts tancats de 2 i 2 (caselles i valors potencials):
    - Suposant que la casella interceptada és de les que estan ocupades: 20
    - Suposant que la casella interceptada té aquests valors potencials: 8

Cal aclarir, però, que totes aquestes diferències tenen un caràcter de modesta referència orientativa, en la mesura que s'obtenen de considerar els membres de les expressions anteriors, que tenen el caràcter de màxims (¿per què us pensaveu, si no, que en el paràgraf que precedia les demostracions teníem cura a dir "serà no superior a"?), i podria ser que oscil·lessin amunt o avall, tot i ser altament improbable que les oscil·lacions arribessin a invertir el signe de la diferència. Molts ja ho haureu vist així des del començament de l'argumentació, però tampoc estarà de més recordar en quin context ens movem, per fer brillar l'evidència: tenint presents les caselles ocupades exteriors a la fila considerada (suposàvem que a totes les files i columnes n'hi hauria una mateixa quantitat, $\mathbf{9-(m+n)}$), és força probable que el nombre de permutacions que efectivament es poden formar assignant valors a les caselles lliures sigui inferior al valor que hem anotat en primer lloc (dels tres, $\mathbf{(m+n)!}$, $\mathbf{m\times(m+n-1)!}$ i $\mathbf{n\times(m+n-1)!}$, el que pertoqui segons el cas) perquè, almenys en les exteriors a la nostra columna protagonista, el nombre de valors admissibles en cada casella pot ser inferior a $\mathbf{(m+n)}$, però és igualment probable que el nombre de permutacions que es poden formar assignant $\mathbf{m}$ valors a $\mathbf{m}$ caselles i $\mathbf{n}$ valors a $\mathbf{n}$ caselles, en cadascun dels conjunts tancats de la columna protagonista que abans havia estat fila, siguin inferiors a $\mathbf{m!}$ i a $\mathbf{n!}$, respectivament, podent arribar a 2 (correspondria al cas en què cadascun dels candidats definitoris del conjunt tancat només aparegués dos cops en total i cada casella només en tingués dos) i, per tant, a 4 el producte, perquè no totes les caselles del primer conjunt admetran els $\mathbf{m}$ valors ni totes les del segon admetran els $\mathbf{n}$ valors definitoris.

Tanmateix, fins i tot acceptant amb indulgència que les dues pàgines precedents no passin de ser una argumentació feble, purament probabilística, ¿no us ha quedat la sensació que s'us escamotejava alguna cosa? Es volia arribar a la conclusió que, quan tenim l'escaquer 9×9 emplenat més o menys uniformement i hi ha una línia en què les caselles lliures s'organitzen en subconjunts tancats de valors candidats, l'exploració serà més curta (i ràpida) si aquesta alineació és una fila que si és una columna, atès que **COMPLET-9*9** < **RE-MEMBER** genera una exploració de caselles per files, saltant de cada fila a la següent. Però a l'hora de comparar les dues situacions ens hem limitat a comparar dues alineacions: la dels subconjunts amb una qualsevol de les ortogonals. És que no hi pinten res la resta d'alineacions? És que les línies paral·leles a la dels subconjunts són intercanviables amb les ortogonals? No pas pel que fa al nombre de combinacions amb els diferents valors que poden adoptar les seves caselles lliures (que, això sí, consideràvem en igual nombre en ambdós casos), que en les **8** del primer grup són no superiors a $(m + n)!$ mentre que en **m** línies del segon són no superiors a $m \times (m + n - 1)!$, en **n** línies són no superiors a $n \times (m + n - 1)!$ i només en les $9 - (m + n)$ línies restants són també no superiors a $(m + n)!$. Passa, però, que fer una avaluació integral de les combinacions possibles en un i altre cas és inviable, i no pas perquè haguéssim de considerar 9 hipòtesis diferents (la presència dels subconjunts cada cop en una de les 9 files), que també, sinó perquè el problema ja transcendeix de la comparació de valors (exactes o purament delimitadors, com els màxims que estavem utilitzant) obtinguts d'expressions algebraiques, i caldria entrar de ple en el territori dels algoritmes que només rutllen amb el concurs de supercomputació: per entendre'ns, estem parlant d'embarcar-nos en treballs de l'envergadura del de B. Felgenhauer i F. Jarvis citat en el capítol *Tornem a començar: de la solució al problema*. Pensar en qualsevol altra cosa en l'àmbit de l'escaquer 9×9 en conjunt, tenint en compte la influència de cada fila sobre les demés (ni que fos prescindint de restriccions a nivell de requadre 3×3 i limitant-les a les de columna), segueix quedant fora de les nostres possibilitats. I baixar més el llistó en l'àmbit 9×9, ignorant aquesta influència, suposa obrir la porta a conclusions encara més discutibles que les que acceptàvem a contracor descontextualitzant fila i columna. Per exemple, admetent la simplificació barroera de prescindir no sols del condicionament dels valors que anem assignant a les caselles d'una fila sobre les següents sinó també dels de les $9 - (m + n)$ caselles ja ocupades en cada fila i columna quan ens posem a especular, la relació comparativa entre el nombre de possibilitats de les dues configuracions vistes, circumscrita a una fila i que era més gran quan l'alineació integrada per subconjunts tancats passava a columna per transposició de l'escaquer, s'invertirà (la presència del signe **<**, que no és tan evident com ho abans era el signe **>**, la justificarem de seguida).

Si, després de la transposició, tenim **m** files en què el nombre de permutacions de valors és (no repetirem allò de "no superior a") $m \times (m + n - 1)!$, **n** files en què és $n \times (m + n - 1)!$ i $9 - (m + n)$ files en què és $(m + n)!$, hauria de ser:

$$[m \times (m + n - 1)!]^m \times [n \times (m + n - 1)!]^n \times (m + n)!^{[9-(m+n)]} < m!\, n!\, (m + n)!^8$$

Si ara dividim tots dos membres per $(m + n)!^{[9-(m+n)]}$, obtindrem:

$$m^m \times n^n \times (m + n - 1)!^{(m+n)} < m!\, n!\, (m + n)!^{(m+n-1)}$$

El **<** d'ara no té un caràcter tan ampli com el **>** d'abans (que només estava fitat inferiorment, amb $m > 1$ i $n > 1$), i únicament n'hem comprovat la pertinència amb els valors que fan al cas: 7 i 2, 6 i 3 o 5 i 4 (9 caselles lliures); 6 i 2, 5 i 3 o 4 i 4 (8); 5 i 2 o 4 i 3 (7); 4 i 2 o 3 i 3 (6); 3 i 2 (9), i 2 i 2 (9).

Per donar credibilitat a un despropòsit com aquest (considerar que la potència a^n del nombre **a** de possibilitats d'emplenar una fila té a veure amb el de **n** files, per bé que volguem disfressar la inconsistència de l'afirmació dient que comparem dos màxims i recorrent a l'habitual "no superior a", equival a utilitzar el valor $9!^9 = 1{,}09111 \times 10^{50}$ en lloc del $6{,}67090 \times 10^{21}$ calculat per els citats Felgenhauer i Jarvis), més val quedar-se amb l'anterior, que al cap i a la fi ens aboca a unes conclusions coincidents amb els resultats experimentals. I estarà carregat de raó qui recrimini a l'autor haver fet aquesta digressió tan llarga per acabar amb una petició de principi. Així que serà millor abandonar discretament un terreny en què mai no hauríem hagut d'entrar, i recuperar el to de modèstia que millor li escau a un treball com aquest: assumir que juguem amb una hipòtesi plausible, abonada per uns quants exemples i que mantindrem mentre cap contraexemple no ens l'ensorri, i sobretot no perdre el fil de d'una argumentació que tant ens havia costat tancar.

L'altra constatació que ens havia desconcertat i ens havia fet perdre cert temps, a la cerca d'alguna causa addicional amb què poder estrènyer el cercle d'aspirants al tractament corrector (inhibir la transposició prèvia o practicar-ne una altra), era que no sempre l'existència de subconjunts disjunts de candidats completant una columna (o el seu predomini numèric sobre els que completaven files) entorpia la proclamació de candidats, o només n'afectava alguns. Però no ens ha de preocupar massa no trobar-la i aplicar preventivament el tractament (com qui administra una vacuna a una població de risc molt àmplia) a totes les situacions que responen a l'unic perfil identificat, després del cas que seguirem amb detall. Al cap i a la fi transposar dues matrius 9×9 no costa gens, tirar pel dret també ho hem fet en l'última incidència i ha estat el criteri habitual en els subcapítols precedents.

Havíem quedat en que les files constituïdes per conjunts disjunts de candidats fan de dreceres que s'agraeixen en un context (de versions **RE-MEMBER** progressivament sofisticades) on els retrocessos cada cop surten més cars, mentre que els mateixos conjunts gairebé no fan sentir cap influència bona quan s'organitzen per columnes. Això també passa quan el nombre de columnes constituïdes per aquests conjunts és superior al de files, cosa que succeïa en el primer del tres exemples considerats: la depuració tindrà uns efectes molt minsos en l'escurçament de les trajectòries. En aquests casos pot ser que la resolució de les exploracions sigui penosa malgrat que l'amplitud de les giragonses no sigui excessiva: de fet, giragonses d'amplitud semblant no produïen demores tan aparatoses entre les primeres versions, perquè la durada de les recursions **RE-MEMBER** (les que proliferaven quan **INI** = **nil**) era força menor. Però no sempre és així ni passa igual amb tots els valors candidats a una casella, com ha quedat palès en els llistats que hem ofert al lector unes pàgines enrera, cosa que ha pogut provocar certa desorientació perquè els conjunts tancats seguien al mateix lloc, amb un candidat o un altre. Però, de fet, la raó d'aquest comportament arbitrari en la innocuïtat o no de la resposta no és pas res que no haguem vist abans, tot i que ja ens en havíem oblidat: que ho sigui o no, dependrà del millor o pitjor joc que tingui cada candidat, i per copsar això (que la sempre inevitable presència del atzar pot traduir en resultats dispars) n'hi haura prou a considerar la sort ben diversa que corren els veredictes d'aptitud dels candidats **6** i **7** a la casella 7,8 en l'últim dels tres exemples. A l'esquerra, tenim el camí triomfal en què es converteix l'exploració del candidat **6** a 8,1 (que correspon a l'esmentada 7,8 de l'escaquer original, abans que el dispositiu "optimitzador" el transposés, i permutés alguns requadres 9×3 i algunes files), seguint la dinàmica NATURMÍN sense haver de donar ni un pas enrera, igual que passa amb el candidat **9**. Però, en comptes de dedicar espai i temps a aquest altre, serà més profitós fer un seguiment de la representació per tal que ningú, a la vista d'assignacions com 6 a 5,3 i 1 a 6,3, per exemple, sospiti que una afirmació tan rotunda com "ni un pas enrera" és exageració, malgrat respondre als llistats de retrocessos. És l'acció del dispositiu tapaforats alló que més contribueix a evitar giragonses, i per tal que la descripció de la seqüència d'emplenaments no sigui tan farragosa convindrem en subratllar els que es produeixen automàticament: 7 (4,1), 8 (5,1), 9 (9,1), 4 (1,2), 5 (2,2), 6 (3,2), 9 (4,2), 2 (5,2), 3 (6,2), 7 (7,2), 8 (9,2), 7 (1,3), 8 (2,3), 9 (3,3), 5 (4,3), 1 (6,3), 6 (5,3), 2 (7,3), 3 (8,3), 4 (9,3), 3 (1,4), 6 (2,4), 2 (3,4), 5 (9,4), 7 (6,4), 9 (5,6), 7 (5,7), 6 (9,7), 8 (4,7), 9 (6,7), 8 (7,4), 9 (8,4), 8 (1,5), 6 (6,5), 1 (7,5), 9 (2,5), 5 (1,6), 8 (6,6), 1 (2,6), 6 (7,6), 3 (9,6), 4 (3,5), 7 (3,6), 4 (8,6), 2 (8,5), 7 (9,5), 6 (1,8), 4 (4,8), 7 (2,8), 8 (8,8), 5 (3,8), 2 (6,8), 1 (9,8), 3 (5,8), 9 (7,8), 9 (1,9), 4 (2,9), 8 (3,9), 6 (4,9), 1 (5,9), 5 (6,9), 3 (7,9), 7 (8,9) i 2 (9,9).

9 4 8	6 1 5	3 7 2	0 0 0	0 0 0	0 0 0	9 7 8	4 3 5	6 2 1
6 7 5	4 3 2	9 8 1	0 0 0	0 0 0	0 0 0	6 4 5	8 2 1	3 9 7
1 2 3	8 7 9	**4 5** 6	**1 2 3**	0 X 0	**4 5** 7	**1 2 3**	9 7 6	**4 5** 8
5 1 7	**2** 9 8	6 4 3	0 0 0	**2** 6 0	0 0 0	5 9 7	**2** 6 8	1 3 4
8 9 4	**3 5** 6	1 2 7	0 0 0	**3 5** 0	0 0 0	8 1 4	**3 5** 9	7 6 2
3 6 2	**1 4** 7	8 9 5	3 6 2	**1 4** 7	0 0 5	3 6 2	**1 4** 7	9 8 5
7 8 9	5 6 1	2 3 4	7 8 9	5 1 3	2 4 6	7 8 9	5 1 3	2 4 6
4 5 6	9 2 3	7 **1** 8	4 5 6	7 9 2	3 **1** 8	4 5 6	7 9 2	8 **1** 3
2 3 1	7 8 **4**	**5 6** 9	**2 3 1**	6 8 **4**	**5 7** 9	**2 3 1**	6 8 **4**	**5 7** 9

Però amb el candidat **7** la cosa no anirà tan bé, perquè l'exploració ja es troba un entrebanc a la casella 7,2 i anàlogament passarà amb el candidat **8**. Ens limitarem al primer però en compensació descriurem la incidència amb cert detall, com abans: en el centre representem la primera embranzida, en què la primera assignació a la casella esmentada es fa amb el candidat 3, queda aturada en emplenar rutinàriament 3,4 amb el candidat 2, seguir amb els emplenaments automàtics 9,4 (5), 6,4 (7) i 5,6 (6), i quedar la casella 5,7 (X) sense valors admissibles, raó per la qual cal emprendre la retirada, retrocedint fins la posició 7,2 (replegament que no serà un retorn directe a casa, sinó que estarà quallat d'intents infructuosos de proseguir l'avanç amb els candidats 5, 7 i 8 a la casella 3,4; 7 i 9 a 2,4; 5, 6, 8 i 9 a 1,4; 6 a 8,3, 6 a 7,3; 3 a 5,3; 9 a 2,3, i 8 i 9 a 1,3) per provar fortuna amb l'altre candidat, el 8; l'emplenament total de l'escaquer amb aquest candidat, ja sense més incidències, és el que mostra la representació dreta. Ens limitarem a descriure l'embranzida inicial avortada i la recuperació a partir de l'assignació del segon candidat a la casella 7,2, per no fer encara més difícil el seguiment (les ziga-zagues entre la posició 3,4 i l'anterior us les estalviem, tot i que la seva longitud seria la prova material de la demora que es produeix en aquest cas, però seria difícil pair una seqüència detallada amb 4.534 retrocessos i un nombre igual de passos endavant): 6 (4,1), 8 (5,1), 9 (9,1), 4 (1,2), 5 (2,2), 6 (3,2), 7 (4,2), 9 (5,2), 5 (4,3), 2 (6,2), 3 (7,2), 8 (9,2), 1 (8,2), 7 (1,3), 8 (2,3), 9 (3,3), 1 (5,3), 3 (6,3), 2 (7,3), 4 (8,3), 6 (9,3), 7 (9,7), 3 (1,4), 6 (2,4), 2 (3,4), 5 (9,4), 7 (6,4), 6 (5,6), X (5,7) i replegament fins a (7,2); 8 (7,2), 3 (9,2), 7 (1,3), 8 (2,3), 9 (3,3), 1 (5,3), 3 (6,3), 2 (7,3), 4 (8,3), 6 (9,3), 3 (1,4), 6 (2,4), 2 (3,4), 5 (9,4), 7 (6,4), 9 (7,4), 8 (8,4), 6 (5,6), 3 (8,6), 7 (5,7), 8 (9,7), 9 (4,7), 6 (6,7), 8 (1,5), 8 (6,6), 1 (2,5), 4 (3,5), 4 (9,6), 1 (7,6), 7 (7,5), 2 (9,5), 6 (8,5), 5 (1,6), 7 (3,6), 9 (2,6), 6 (1,8), 3 (7,8), 2 (5,8), 9 (8,8), 9 (1,9), 3 (5,9), 6 (7,9), 2 (8,9), 4 (2,8), 8 (4,8), 5 (3,8), 1 (6,8), 7 (9,8), 7 (2,9), 8 (3,9), 4 (4,9), 5 (6,9) i 1 (9,9).

En definitiva, i encara que no quedi elegant, una forma entenedora de presentar-ho és afirmar (en gallec) que als nostres subconjunts disjunts organitzats en columna els passa *como os pementos de Padrón, que uns pican e outros non*, de manera que, si a algú no li plau el picant, millor que no en consumeixi d'aquesta procedència.

Abans de traduir a codi el dispositiu de prevenció, cal prendre una decisió sobre una alternativa que havíem apuntat: quan detectem la presència de subconjunts que s'organitzen en columna i afecten més caselles que les eventualment afectades per subconjunts organitzats en fila, ¿és millor interrompre l'exploració en curs i reiniciar **MASCARA** per evitar la transposició automàtica prèvia (o forçar-la, si no n'hi hagués hagut), o efectuar una transposició a posteriori que inverteixi aquest balanç i només es produeixi amb els candidats en què calgui? És clar que la segona és l'opció correcta, no només per raons d'eficiència sinó per impedir que el remei per un candidat pugui representar l'esguerro per un altre. L'acció tindrà lloc a **RE-MEMBER**, al final de l'etapa **INI**cial, i estarà condicionada a **(> COL FIL)**, on **COL** i **FIL** són variables locals en la funció i representen el nombre de caselles implicades en els dos tipus de subconjunts. No les inicialitzem a **RE-MEMBER** sinó prèviament a cada execució d'**ACTUALITZA-PLUS** ordenada des d'**ACTUALITZA**, quan s'hi accedeix des de **RE-MEMBER**, i ho fem així perquè tal com ho tenim muntat serà més fàcil fer el recompte en cada execució d'**ACTUALITZA-PLUS** que portar un inventari de les caselles depurades i afegir només les dels nous subconjunts identificats. A més de les funcions esmentades, també reproduïm **EXPLORA-9+9+5** (que ara inclou l'argument **FIL/COL**, que valdrà **1**, **2** o **nil** segons que els accessos corresponguin a files, columnes o requadres 3×3) i **EXPL-SUBCONJ** on es fa el recompte de **FIL** i **COL**. Que les caselles comuns a un conjunt-fila i un conjunt-columna es comptin dos cops no altera el signe del balanç, ni us ha d'estranyar que s'ignorin els subconjunts organitzats en requadres 3×3: si quan, en el primer dels tres casos considerats, parlàvem de subconjunts que abastaven un requadre 9×3 (transposat en requadre 3×9) algú ho assimilava a tres files (tres columnes), s'oblidava que en la pàgina 334 només havíem identificat com a conjunts-fila la 5ª i 6ª; la 4ª fila no ho era pas. Aquí teniu les cinc funcions amb les petites modificacions comentades:

```
(defun EXPL-SUBCONJ (/ L OK)
 (setq K 1)
 (foreach Ñ (nth (- (length F-L) 4) LLISTES)
   (if (not (or PLUS FORA))
       (progn
         (setq K (1+ K))
```

```
          (foreach F Ñ
            (if (not (or PLUS FORA))
                (progn
                  (setq L ())
                  (if (< (1+ K) (length F-L))
                      (progn
                        (PRESENTS)
                        (if (= (length L) K)
                            (progn
                              (setq P-DEP () OK ())
                              (foreach E F (setq P-DEP (cons (nth E F-9) P-DEP)))
                              (foreach M L
                                (if (not OK)
                                    (mapcar '(lambda (E-9 E-L)
                                               (if (and (not OK)
                                                        (not (member E-9 P-DEP))
                                                        (member M E-L))
                                                   (setq OK T L-DEP L PLUS T)))
                                            F-9 F-L)))
                              (if (and FIL/COL (not OK))
                                  (if (= FIL/COL 1)
                                      (setq FIL (+ FIL K))
                                      (setq COL (+ COL K))))))))
                  (if (and (not FORA) (> K 2))
                      (progn
                        (if (not L) (PRESENTS))
                        (if (< (length L) K) (setq FORA T)))))))))))

(defun EXPLORA-9+9+5 (FIL/COL / NN MN F-9 F-L)
  (foreach Ñ L1A9 (if (not (member Ñ F-9*9)) (setq NN (cons Ñ NN))))
  (if (> (length NN) 0)
      (progn
        (setq NN (reverse NN) MN (last NN))
        (foreach Ñ NN
          (if (not PLUS)
              (progn
                (setq K 0)
                (if (and INI (= Ñ MN)) (setq F-9 () F-L ()))
                (mapcar '(lambda (E-9*9 E-LL3)
                           (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
                           (if (and INI E-LL3 (= Ñ MN))
                               (progn
                                 (setq I ())
                                 (foreach M E-LL3 (if M (setq I (cons M I))))
                                 (setq F-9 (cons E-9*9 F-9) F-L (cons I F-L)))))
                        F-9*9 F-LL3)
                (if (= K 1)
                    (PLUS-N->P J)
                    (if (= K 0)
                        (setq FORA T)
                        (if (and INI (= Ñ MN) (> (length F-L) 3))
                            (EXPL-SUBCONJ)))))))))))

(defun ACTUALITZA-PLUS (/ I L M Ñ X1 Y1 5P R-9 R-L ORIG)
  (setq FORA ())
  (mapcar '(lambda (9*9 LL3)
             (setq ORIG (not ORIG))
             (if (not (or PLUS FORA))
                 (mapcar '(lambda (F-9*9 F-LL3)
                            (if (and (not (or PLUS FORA)) ORIG)
                                (mapcar '(lambda (E-9*9 E-LLL)
                                           (if (and (not (or PLUS FORA)) E-LLL)
                                               (progn
                                                 (setq K 0)
                                                 (foreach E E-LLL
                                                   (if (and (< K 2) E)
                                                       (setq Ñ E K (1+ K))))
                                                 (if (= K 1)
```

```
                                                    (PLUS-N->P E-9*9)
                                                    (if (= K 0)
                                                        (setq FORA T))))))
                                        F-9*9 F-LL3))
                                (if (not (or PLUS FORA))
                                    (EXPLORA-9+9+5 (if ORIG 1 2))))
                             9*9 LL3)))
          (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL)))
  (if (not (or PLUS FORA))
      (progn
        (setq X1 (* (/ X 3) 3) Y1 (* (/ Y 3) 3) ORIG ())
        (foreach K '(6 3 0) (if (/= K Y1) (setq 5P (cons (list X1 K) 5P))))
        (foreach K '(6 3 0) (setq 5P (cons (list K Y1) 5P)))
        (repeat (if INI 2 1)
          (setq ORIG (not ORIG))
          (if (not (or PLUS FORA))
              (foreach Y '(0 3 6)
                (foreach X '(0 3 6)
                  (if (and ORIG (not (or PLUS FORA)) (member (list X Y) 5P))
                      (progn
                        (setq XY (list (1+ X) (1+ Y))
                              R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              5P (subst (cons (list X Y) (list R-9 R-L))
                                                         (list X Y) 5P))
                        (mapcar '(lambda (F-9*9 F-LL3)
                                   (if (not (or PLUS FORA))
                                       (EXPLORA-9+9+5 ())))
                                (list (append (car R-9) (cadr R-9) (last R-9)))
                                (list (append (car R-L) (cadr R-L) (last R-L))))))
                  (if (not ORIG)
                      (progn
                        (if (setq R-L (assoc (list X Y) 5P))
                            (setq R-9 (cadr R-L) R-L (caddr R-L))
                            (setq R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))
                                  R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))))
                        (setq K 0)
                        (foreach F R-9
                          (foreach E F (if (atom E) (setq K (1+ K)))))
                        (if (< K 9) (TERCETS R-9 R-L))
                        (setq N/R-L (3*3/R-L N/R-L)))))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y XY L PLUS P-DEP L-DEP)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-9*9 (ACT-9*9 E (list J K) A-9*9))
                  (command "ESPACIOM" "TEXTO" "MC" (list J K) 0.5 0 (itoa E)))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P) FIL 0 COL 0)
             (if L-DEP
                 (setq A-LLL (ACT-DEPURA) DEP T L-DEP () XY ())
                 (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
                       A-LLL (ACT-LLL X Y A-LLL) XY ()))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))
```

```
(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P FIL COL)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if (or TT DEP)
                  (progn
                    (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)))
                    (setq 2R (list N (REORD R-LLL) (REORD REPLUS))
                          TN/R-L (if TT
                                     (subst 2R (cons N TT) TN/R-L)
                                     (if TN/R-L
                                         (cons 2R TN/R-L)
                                         (list 2R))))))
              (if (> COL FIL)
                  (setq R-9*9 (TRANSPOSAR R-9*9)
                        R-LLL (TRANSPOSAR R-LLL) K ()
                        R-9*9 (foreach F (reverse R-9*9)
                                (setq J ())
                                (foreach E (reverse F)
                                  (setq J (cons (if (atom E) E (reverse E)) J)))
                                (setq K (cons J K)))))))))
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (and (not OK) (member E R-N))
              (progn
                (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                (if (not FORA)
                    (if (equal (cadr 2R) NIL*NIL)
                        (setq OK T)
                        (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)
```

Però no cantem victòria, perquè n'hi haurà prou a furgar una mica en l'últim cas representat uns deu fulls enrera, despullant-lo d'alguns valors (com **4** a 7,7, **5** a 8,7 i **7** a 8,1) i reculant fins a l'activació de la casella 3,7 per situar-nos en una etapa prèvia a la problemàtica dels conjunts tancats de candidats, per poder clamar en castellà allò de *¡Mi gozo en un pozo!* De fet, quan en els assajos sobre aquell cas seguíem un ordre d'emplenament ascendent, deixant la casella 8,1 per al final, ja ens trobàvem amb dues demores, corresponents a 3,7 (candidat **5**, amb una espera de més de mig minut) i 8,7 (candidat **5**, amb un minut), per bé que haguéssim posat en el punt de mira l'esmentat final de seqüència per ser doblement enutjós (candidats **7** i **8**, amb un minut cadascun). Doncs bé: amb l'última versió hem pogut normalitzar la resposta a 8,7 i 8,1, però no a 3,7; així doncs, tornem-hi que no ha estat res! La representació de l'esquerra reflecteix aquesta situació, la de la dreta ens mostra l'escaquer 9×9 emplenat seguint la dinàmica NATURMÍN compatible amb la presència del valor **5** en aquesta casella, i en el centre oferim la primera embranzida fins que **ACTUALITZA-PLUS** detecta l'absència de candidats en la posició 6,7, primera ensopegada que marca l'inici d'una retirada tàctica que pot explicar el fet que la batalla trigui tant a guanyar-se:

0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	9 4 8	7 1 5	6 3 2
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	6 7 3	9 8 2	1 5 4
1 2 5	0 0 0	0 0 0	**1 2 5**	0 0 X	0 0 0	**1 2 5**	4 6 3	9 7 8
0 0 0	**2** 0 0	0 0 0	5 0 0	**2** 9 8	0 0 0	5 1 7	**2** 9 8	4 6 3
0 0 0	**3 5** 0	0 0 0	8 9 0	**3 5** 6	0 0 0	8 9 4	**3 5** 6	7 2 1
0 0 0	**1 4** 0	0 0 0	3 6 2	**1 4** 7	8 9 5	3 6 2	**1 4** 7	8 9 5
0 0 0	0 0 0	0 0 0	7 8 9	5 1 3	2 4 6	7 8 9	5 3 1	2 4 6
0 0 0	0 0 0	0 **1** 0	4 5 6	8 2 9	3 **1** 7	4 5 6	8 2 9	3 **1** 7
2 3 1	0 0 **4**	**5** 0 0	**2 3 1**	6 7 **4**	**5** 8 9	**2 3 1**	6 7 **4**	**5** 8 9

Abans d'entrar en l'estudi de les causes concretes, però, aclarim que en el moment de produir-se la incidència l'última casella ocupada recursivament per **RE-MEMBER** era 1,5 (8, valor que per expresar aquesta circumstància ha estat subratllat), perquè 1,6 (5) i 5,6 (9) són forats emplenats automàticament amb anterioritat, i 6,5 (6) i 6,6 (8) ho són just després de 1,5 (8). Dit això, comparem els estats reflectits en segon i tercer lloc i fixem-nos que la primera diferència apareix a la tercera fila i sembla nímia: els valors 1 i 3 estan permutats en les posicions 5,3 i 6,3. Doncs la insignificància només és aparent, perquè és en aquest punt, o més exactament, en la lenta regressió fins aquesta posició, plagada de giragonses, on rau la llarga espera. Si més no, el remei *ad hoc* que se'ns va ocórrer semblava corroborar-ho perquè funcionava: constatat que 6,7 s'acabava convertint en un cul-de-sac quan anàvem explorant l'escaquer a partir de la primera casella desocupada després d'emplenar virtualment 3,7 (**5**), invertiríem la situació i començaríem amb aquesta posició, seguint després l'ordre habitual a partir de la primera lliure. És clar que això només seria eficaç limitant-ho al primer atzucac localitzat (ja n'hi havia prou d'abusar tant de "cul-de-sac", tenint un altra mot ben nostrat), o al segon o al tercer, però no ho podíem generalitzar perquè seria pitjor el remei que la malaltia. Vet aquí com han canviat les coses en dos subcapítols!: en *Etapa prèvia: crear una solució, III*, havíem albirat la conveniència de detectar quines caselles s'havien quedat sense valors admissibles en primera instància, per tal d'abandonar la recursió de **RE-MEMBER** que ens hi havia dut; ara, no n'hi ha prou amb això sinó que, almenys el primer cop, proposem alterar l'ordre d'exploració. Però cal no treure conclusions precipitades d'aquesta altra volta de rosca, perquè amb els comentaris que s'havien fet unes pàgines enrera a propòsit de l'elevació del cost en temps de cada retrocés (per la complicació del programa, al seu torn conseqüència de la necessitat de fer front a situacions més alambinades), en cada nova versió del programa, algú podria pensar que hem caigut en un cercle viciós on cada nova cautela llasta la rapidesa d'execució, de manera que certes situacions que en versions mes primitives resultaven innòqües ara es tradueixen en esperes inacceptables, per acumulació de comprovacions innecesàries. Res d'això: sí que és veritat que cada retrocés d'ara triga més que un d'abans, però també ho és que de mitjana els veredictes es resolen amb trajectòries més curtes, i no només el segon factor compensa el primer sinó que en general ho fa amb escreix perquè els temps s'han escurçat; no és una simple presumpció sinó que s'ha verificat executant els exemples del subcapítol esmentat amb l'última versió; també s'ha comprovat que les demores que considerem ara també s'haurien produït (gairebé sempre augmentades, i de vegades molt) utilitzant aquelles versions.

Però anem a veure com ens ho hem muntat per interrompre l'exploració en curs i reprendre-la activant virtualment la casella-atzucac immediatament després de la casella activada per l'usuari (atenció: només amb el valor que provoca l'atzucac). La necessitat d'escometre el canvi vindrà donada per la variable **CANVI**, local en **MASCARA** com **PRIMER**, que controlarà que això només es produeixi la primera vegada. La detecció de l'atzucac, que tenia lloc en **ACTUALITZA-PLUS** des de feia ja unes quantes versions, mitjançant el dispositiu

```
  .................
  (setq K 0)
  (foreach E E-LLL
    (if (and (< K 2) E) (setq Ñ E K (1+ K))))
  (if (= K 1)
      (PLUS-N->P E-9*9)
      (if (= K 0) (setq FORA T)))...
```

l'hem convertida en

```
  .................
  (setq K 0)
  (foreach E E-LLL
    (if (and (< K 2) E) (setq Ñ E K (1+ K))))
  (if (= K 1)
      (PLUS-N->P E-9*9)
      (if (= K 0)
          (progn
            (if (and PRIMER (not INI))
                (setq CAUSANT (car CAMI) NOLLL E-9*9 PRIMER ()))
            (setq FORA T))))...
```

on la variable **CAUSANT** representa la casella l'emplenament virtual de la qual ha causat l'atzucac (entre l'un i l'altra hi ha pogut haver emplenaments automàtics) i **NOLLL** la posició on s'ha produït. Pel que fa a **CAMI**, és una llista amb totes les

caselles emplenades, començant per la que explorem, excloent-hi els forats tapats automàticament i que no ens hem molestat a invertir (el primer element és l'última casella emplenada). Però serà necessari veure com ha quedat **RE-MEMBER** per entendre el procés de formació de la llista i com s'actualitza durant els retrocessos:

```
(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P FIL COL)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              ..................................................
              (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P))))))
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (if CANVI
            (setq R-P NOLLL NOLLL () CANVI ())
            (setq CAMI (cons R-P CAMI)))
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (and (not OK) (not CANVI) (member E R-N))
              (progn
                (if (or PRIMER NOLLL)
                    (progn
                      (setq CAMI (member R-P CAMI))
                      (if (and NOLLL (not (member CAUSANT CAMI)))
                          (setq CANVI T))))
                (if (not CANVI)
                    (progn
                      (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                      (if (not FORA)
                          (if (equal (cadr 2R) NIL*NIL)
                              (setq OK T)
                              (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)
```

Com podeu veure, en la primera execució es va fent **CAMI**, exclusivament amb les posicions emplenades autorecursivament, ignorant els forats tapats automàticament i escurçant-la cada vegada que hi ha una regressió: **(setq CAMI (member R-P CAMI))** serveix per actualitzar **CAMI** tant en el cas que el candidat que ha donat lloc a l'atzucac sigui substituit per un altre, com en el cas que no n'hi hagin més i es produeixi un retrocés; ja es comprèn que és aquesta darrera la circumstància que ens interessa enregistrar, i per això l'activació de **CANVI** estarà condicionada a **(not (member CAUSANT CAMI))**. Com veureu de seguida, quan s'esdevé això s'nterromp l'execució i n'hi haurà una segona caracteritzada pel fet que ja deixem de banda **CAMI** i retornem a la rutina de les versions anteriors, tret del fet que la primera casella a explorar, **R-P**, no és la primera desocupada (localitzada a **COMPLET-9*9**) sinó la casella-atzucac **NOLLL**. Pel que fa a la funció **MASCARA**, que ara incorpora les variables **RR-9*9**, **RR-LLL**, **PRIMER**, **CAUSANT**, **CAMI**, **CANVI** i **RE-MEM**, el dispositiu

```
  ..............
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn ...)))...
```

s'ha convertit en

```
  ..............
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                 (progn
                   (setq RE-MEM (RE-MEMBER 0-**9*9** 0-LLL))
                   (if CANVI (setq RE-MEM (RE-MEMBER RR-9*9 RR-LLL)))
                   RE-MEM)))
        (progn ...)))...
```

mostrant-nos com, quan s'ha activat **CANVI**, torna a executar-se **RE-MEMBER** amb els valors **R-9*9 R-LLL** de la primera part (**INI** = **T**) de l'execució inicial, transferits a **RR-9*9 RR-LLL**, i passant directament a la segona part (perquè ara és **INI** = **nil**).

Recapitulant, direm que **CANVI** únicament s'activa amb la primera regressió, sempre que retocedeixi fins a una posició anterior a la casella **CAUSANT**, i a aquests efectes cal precisar que només hem considerat les regressions degudes a atzucacs puntuals, és a dir, a caselles buides i desproveïdes de qualsevol valor potencial (l'element de **R-LLL** que les representa és '**(() () () () () () () () ())**), no les degudes a files, columnes o requadres 3×3 carents d'un valor 1... 9 determinat (entre els assignats i els admissibles) i localitzats per la funció **EXPLORA-9+9+5**.

Però no podrem recuperar per a l'ocasió la frase lapidària de Neil Armstrong i dir que això ha estat un petit pas per a un home però un gran salt per a la Humanitat. Més aviat ha estat un pas petitet en relació a l'esforç desenvolupat, i veureu que encara haurem d'avançar a petits passets fins arribar a treure'n l'entrellat. Per començar, tornem a l'exemple que dúiem entre mans, celebrem l'éxit assolit entrant el valor **5** a 3,7, i activem la casella 2,2: el candidat **8** trigarà més de mig minut a aparèixer i, si després de l'espera encara ens ve de gust emplenar ordenadament les caselles desocupades 4,0, 5,0, 8,0, 9,0, 1,2 i 3,2, ens trobarem amb demores semblants amb els primers candidats vàlids (sí, senyor: amb els compatibles amb la dinàmica NATURMÍN de l'estat precedent, que segueixen sent-ho amb un **8** a 2,2), que són **6**, **7**, **8**, **9** (emplenament automàtic), **4** i **6**. Limitem-nos, però, a aquest primer emplenament problemàtic, reflectit a l'esquerra, a dalt. A baix, a la dreta, teniu la solució NATURMÍN compatible i entremig les tres primeres embranzides seguida de un retrocés que activa CANVI i que inicia la regressió fins a la casella 5,3 (1): compareu aquesta posició i la següent en l'última representació i en les quatre precedents i, com abans, us adonareu que els valors 1 i 3 hi apareixen permutats. Més exacte seria dir que representen tres situacions consecutives que culminen la primera embranzida: a la segona, l'emplenament virtual de 3,5 amb el candidat 4 desencadena l'automàtic 7,5 (1) (l'altre, 8,5 (2), ja hi era), i la casella 9,5 es queda sense candidats; a la tercera, el candidat 7 pren el relleu, desencadenant els emplenaments automàtics 6,5 (8), 1,6 (8), 3,6 (4), 6,6 (7) i 7,6 (6) (5,6 (9) i 8,5 (2) ja hi eren), i la casella 6,7 és qui queda sense candidats; a la quarta, l'últim candidat compatible amb les ocupacions actuals és 8, que desencadena els emplenaments automàtics 6,5 (7), 1,6 (7), 3,6 (4), 6,6 (8) i 7,6 (6) (recordeu que 5,6 (9) i 8,5 (2) ja hi eren), repetint-se l'atzucac en la mateixa posició, i a la cinquena no és un pas enrera sinó dos (de 3,5 (8) a 1,5 (7)) el que activa **CANVI**.

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 0   0 0 0      1 2 5   0 0 0   0 0 0      1 2 5   0 0 X   0 0 0

0 0 0   2 0 0   0 0 0      0 1 0   2 9 0   0 0 0      8 1 4   2 9 7   6 0 0
0 0 0   3 5 0   0 0 0      6 9 4   3 5 0   1 2 X      6 9 7   3 5 8   0 2 0
0 0 0   1 4 0   0 0 0      3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   7 9 8

0 0 0   0 0 0   0 0 0      5 7 9   8 1 3   2 4 6      5 7 9   8 1 3   2 4 6
0 8 0   0 0 0   0 1 0      4 8 6   5 2 9   3 1 7      4 8 6   5 2 9   3 1 7
2 3 1   0 0 4   5 0 0      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9



0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      9 6 3   4 1 5   8 7 2
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      7 4 8   9 6 2   1 5 3
1 2 5   0 0 X   0 0 0      1 2 5   0 0 0   0 0 0      1 2 5   7 8 3   9 6 4

7 1 4   2 9 8   6 0 0      0 0 0   2 9 0   0 0 0      8 1 4   2 9 7   6 3 5
6 9 8   3 5 7   0 2 0      7 0 0   3 5 0   0 0 0      6 9 7   3 5 8   4 2 1
3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   7 9 8

5 7 9   8 1 3   2 4 6      5 7 9   8 1 3   2 4 6      5 7 9   8 3 1   2 4 6
4 8 6   5 2 9   3 1 7      4 8 6   5 2 9   3 1 7      4 8 6   5 2 9   3 1 7
2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9
```

A més d'activar **CANVI**, aquest tercer atzucac hauria d'haver quedat enregistrat a **NOLLL**, per obviar una llarga i accidentada regressió per intercanviar el contingut de les caselles 5,3 i 6,3, avortant les recursions de **RE-MEMBER** i reiniciant-les

amb 6,7 com a primera casella a explorar virtualment (després de l'activació real de 2,2 (**8**)). Però les coses no han anat així: és el **PRIMER** atzucac qui desactiva aquesta variable, fixant **NOLLL** a 9,5 i **CAUSANT** a 3,5; i encara que el valor 4 és rellevat per 7 i 8, i és el tercer qui provoca el **CANVI**, **NOLLL** no s' actualitza. Esmenar aquesta relliscada, fent que **CANVI** s'activi amb la primera regressió que retocedeixi fins a una posició anterior a la casella **CAUSANT** (en comptes de voler activar-la amb la primera regressió, però només si retocedeix fins a una posició anterior a la casella **CAUSANT**, i a sobre fer-ho malament), no pot ser més senzill: no desactivar **PRIMER** a l'expressió **(setq CAUSANT (car CAMI) NOLLL E-9*9 PRIMER ())** introduïda abans en **ACTUALITZA-PLUS**, i fer-ho prèvia la segona crida a **RE-MEMBER** de **MASCARA**, on la línia **(if CANVI (setq RE-MEM (RE-MEMBER RR-9*9 RR-LLL)))** quedarà **(if CANVI (setq PRIMER () RE-MEM (RE-MEMBER RR-9*9 RR-LLL)))**. Però prengueu-vos-ho amb filosofia, perquè el resultat serà desconcertant: amb l'esmena, el candidat **8** ja no es farà esperar tant; tanmateix, les fins ara inofensives candidatures 6 i 7 arribaran amb una demora de gairebé mig minut cadascuna. Què se'ns està escapant? Veiem-ho baixant als detalls de l'embranzida inicial amb **6** com a candidat a 2,2, i suposant que amb **7** arribaríem a les mateixes conclusions. La primera representació mostra aquest punt de partença, les tretze següents reflecteixen uns aztucacs que corresponen a estats consecutius en la exploració (tret de les progressions que es produeixen entre la 7ª i 8ª, i entre la 10ª i 11ª), i a l'última, com és habitual, us oferim la solució NATURMÍN compatible. Per no fer-nos pesats en la descripció, ja no enumerarem les caselles que, abans de l'últim emplenament ordinari (seguint el patró NATURMÍN) o a conseqüència d'ell, han estat ocupades automàticament per tapar algun forat, com seria el cas de 7,2 (3) en el segon fotograma: el lector ja prou que entendrà que totes les caselles plenes posteriors a la subratllada ho han estat d'aquesta manera. Comencem amb la seqüència de la 2ª a la 7ª representació: aquests atzucacs (a 9,2 i 4,3) se succeeixen ràpidament i són intents inútils de tirar endavant amb els candidats de les caselles 5,2 (2, 3, 8 i 9) i 4,2 (8 i 9).

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 0   0 0 0      1 2 5   0 0 0   0 0 0      1 2 5   0 0 0   0 0 0

0 0 0   2 0 0   0 0 0      0 0 0   2 0 0   0 0 0      0 0 0   2 0 0   0 0 0
0 0 0   3 5 0   0 0 0      0 0 0   3 5 0   0 0 0      0 0 0   3 5 0   0 0 0
0 0 0   1 4 0   0 0 0      0 0 0   1 4 0   0 0 0      0 0 0   1 4 0   0 0 0

0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 6 0   0 0 0   0 1 0      4 6 7   5 2 0   3 1 X      4 6 7   5 3 0   2 1 X
2 3 1   0 0 4   5 0 0      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9



0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 0   0 0 0      1 2 5   0 0 0   0 0 0      1 2 5   0 0 0   0 0 0

0 0 0   2 0 0   0 0 0      0 0 0   2 0 0   0 0 0      0 0 0   2 0 0   0 0 0
0 0 0   3 5 0   0 0 0      0 0 0   3 5 0   0 0 0      0 0 0   3 5 0   0 0 0
0 0 0   1 4 0   0 0 0      0 0 0   1 4 0   0 0 0      0 0 0   1 4 0   0 0 0

0 0 0   X 0 0   0 0 0      0 0 0   X 0 0   0 0 0      0 0 0   X 0 0   0 0 0
4 6 7   5 8 9   0 1 0      4 6 7   5 9 8   0 1 0      4 6 7   8 9 5   0 1 0
2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9



0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 0   0 0 0      1 2 5   0 0 X   0 0 0      1 2 5   0 0 X   0 0 0

0 0 0   2 0 0   0 0 0      8 1 4   2 9 7   6 0 0      0 0 0   2 9 7   0 0 0
0 0 0   3 5 0   0 0 0      6 9 7   3 5 8   0 2 0      7 9 0   3 5 8   0 0 0
0 0 0   1 4 0   0 0 0      3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   7 9 8

0 0 0   X 0 0   0 0 0      5 7 9   8 1 3   2 4 6      5 7 9   8 1 3   2 4 6
4 6 7   9 8 5   0 1 0      4 6 8   5 2 9   3 1 7      4 6 8   5 2 9   3 1 7
2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9
```

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 X   0 0 0      1 2 5   0 0 X   0 0 0      1 2 5   0 0 X   0 0 0

0 0 0   2 9 8   0 0 0      0 0 0   2 9 7   0 0 0      8 1 6   2 9 7   4 0 0
8 9 0   3 5 7   0 0 0      9 1 7   3 5 8   0 0 0      9 4 7   3 5 8   0 0 0
3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   7 9 8

5 7 9   8 1 3   2 4 6      5 7 9   8 1 3   2 4 6      5 7 9   8 1 3   2 4 6
4 6 8   5 2 9   3 1 7      4 6 8   5 2 9   3 1 7      4 6 8   5 2 9   3 1 7
2 3 1   6 7 4   5 0 0      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9



0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      9 8 3   4 6 5   1 7 2
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      7 4 6   9 1 2   8 5 3
1 2 5   0 0 X   0 0 0      1 2 5   0 0 0   0 0 0      1 2 5   7 8 3   9 6 4

0 0 0   2 9 8   0 0 0      0 0 0   2 9 0   0 0 0      8 1 4   2 9 7   6 3 5
9 8 0   3 5 7   0 0 0      0 0 0   3 5 0   0 0 0      6 9 7   3 5 8   4 2 1
3 5 2   1 4 6   7 9 8      3 5 2   1 4 6   9 0 8      3 5 2   1 4 6   7 9 8

5 7 9   8 1 3   2 4 6      5 7 9   8 1 3   2 4 6      5 7 9   8 3 1   2 4 6
4 6 8   5 2 9   3 1 7      4 6 8   5 2 9   3 1 7      4 6 8   5 2 9   3 1 7
2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9      2 3 1   6 7 4   5 8 9
```

Fixeu-vos que, tal com havíem deixat les coses, el retrocés de 5,2 a 4,2 de la 6ª representació activarà el **CANVI:** ¿de debò que val la pena interrompre l'exploració i reprendre-la amb 4,3 com a primera casella a visitar, després de 2,2 (**6**)? Ens ho preguntem perquè en realitat el trànsit de la 2ª a la 7ª és immediat i no deixa al darrera cap diferència en relació a la 15ª representació, cosa que sí que succeeix a la 8ª en què, des del resultat 3,2 (8) de l'últim retrocès se'n va de dret fins a 1,5 (6), deixant enrera una diferència respecte a la solució final: les caselles 5,3 (1) i 6,3 (3), que haurien de permutar els seus valors però que encara no ho hauran fet en arribar a la 14ª representació, punt en què, si hem provat d'inhibir el **CANVI** d'abans, el retrocés de 2,5 (8) a 7,4 (9) activarà un **CANVI** més profitós. ¿Per què aquest canvi sí que ens evitarà que la consecució de l'emplenament total no es faci esperar? Perquè la regressió entre 7,4 (9) i 5,3 (2), que no surt però que ocuparia l'espai entre la representació 14ª i la 15ª, és plena de giragonses i la responsable del retard. I perquè, si ara interrompem l'exploració i la reprenem amb 6,7 (posició de l'atzucac **NOLLL** amb què s'activa **CANVI**) com a primera casella a visitar, el primer candidat (3) farà que el subsegüent emplenament des de 4,1 ja no tingui entrebancs: en arribar a 4,3 (8), **ACTUALITZA-PLUS** taparà automàticament els forats 6,3 (1) i 5,3 (3), i seguirà fins al final sense incidències majors.

Tot això està molt bé, però ¿com podem saber quins atzucacs són inofensius i quins no, per no posar en marxa el dispositiu abreujador abans de trobar el primer del segon grup? Doncs una simple comparació entre l'aspecte dels sis primers atzucacs i el dels sis següents ens donarà la pista: en els primers, l'atzucac **NOLLL** és en el mateix requadre 3×3 o a la mateixa fila que la casella **CAUSANT**, i podem suposar que la diferència oculta amb la solució final es troba a la mateixa unitat modular i això comportarà un reajust immediat, cosa que no passa en els altres. També això és fàcil d'implementar, complicant una mica més en **ACTUALITZA-PLUS** la condició per fixar provisionalment els punts **CAUSANT** i **NOLLL**. En un principi, l'expressió

```
(if (and PRIMER (not INI))
    (setq CAUSANT (car CAMI) NOLLL E-9*9))
```

l'havíem convertit en

```
(if (and PRIMER (not INI) (setq I (car CAMI))
         (not (or (= (car E-9*9) (car I))
                  (= (cadr E-9*9) (cadr I))
                  (and (= (/ (car E-9*9) 3) (/ (car I) 3))
                       (= (/ (cadr E-9*9) 3) (/ (cadr I) 3))))))
    (setq CAUSANT I NOLLL E-9*9))
```

cosa que significava descartar no només els atzucacs esdevinguts dintre del mateix requadre 3×3 o fila que la casella causant, sinó també dins de la mateixa columna. Però a l'hora de treure conclusions dels casos estudiats i de les correccions del codi fins a obtenir en tots ells respostes satisfactòries (força més nombrosos que

els que finalment hem presentat al lector), va semblar que només en els supòsits de pertinença a requadre i fila estava justificada l'exclusió, per la raó (vaga, pero raó al cap i a la fi) adduïda línies enrera, i que no ho estava en el cas de pertinença a columna. Així doncs, i sens perjudici que alguna fallada imprevista aconsellés reincorporar al codi la línia suprimida (no perdem de vista que tot el que venim fent en la segona meitat del present subcapítol podríem dir que respon a una lògica difusa, en la mesura que anem avançant a les palpentes, a partir de constatacions purament empíriques), hem optat per amputar l'expressió precedent, deixant-la en la forma

```
(if (and PRIMER (not INI) (setq I (car CAMI))
         (not (or (= (cadr E-9*9) (cadr I))
                  (and (= (/ (car E-9*9) 3) (/ (car I) 3))
                       (= (/ (cadr E-9*9) 3) (/ (cadr I) 3))))))
    (setq CAUSANT I NOLLL E-9*9))
```

Amb aquestes esmenes quedaran normalitzats tots els veredictes de 2,2, però fent provatures no trigarem a ensopegar una altra vegada: introduïu un **8** en la casella, i seguiu amb 3,2 (**6**), 4,2 (**7**) i 5,3 (**1**); si ara activeu 7,7, el candidat **6** trigarà més d'un quart de minut a aparèixer. La sèrie de nou representacions que teniu tot seguit comença reflectint aquesta situació i, per no perdre el costum, acaba amb la solució NATURMÍN compatible, precedida de la sortida del sisè atzucac, el que activarà **CANVI** i hauria de permetre eludir la penosa regressió des de 6,4 (8) fins a 5,2 (3), ja que la diferència entre la solució final i les set representacions precedents rau en les caselles 5,2 (2) i 7,2 (3), que han de intercanviar valors. Abans d'esbrinar per què **CANVI** no ha produït els efectes desitjats, comentarem que l'atzutzac a 4,3 no dóna lloc a cap **CANVI** perquè la casella causant 1,2 (4) no és ultrapassada per cap retrocés, i que els tres que es produeixen a 6,6 ni tan sols són considerats perquè la casella causant 6,4 se situa en el mateix requadre 3×3.

```
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 0   6 0 0      1 2 5   0 0 0   6 0 0      1 2 5   0 7 8   6 0 3

0 0 0   2 0 0   0 0 0      0 0 0   2 0 0   0 0 0      0 0 0   2 9 X   0 3 5
0 0 0   3 5 0   0 0 0      0 0 0   3 5 0   0 0 0      0 0 0   3 5 7   0 0 0
0 0 0   1 4 0   0 0 0      0 0 0   1 4 0   0 0 0      3 5 2   1 4 6   0 0 7

0 0 0   0 1 0   0 0 0      0 0 0   X 1 0   0 0 0      4 7 9   5 1 3   2 6 8
0 8 6   7 0 0   0 1 0      4 8 6   7 9 5   0 1 0      5 8 6   7 2 9   3 1 4
2 3 1   0 0 4   5 0 0      2 3 1   6 8 4   5 7 9      2 3 1   6 8 4   5 7 9


0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
1 2 5   0 0 8   6 0 0      1 2 5   0 0 7   6 0 0      1 2 5   4 X X   6 X X

0 0 0   2 9 X   0 0 0      0 0 0   2 9 X   0 0 0      0 0 0   2 9 0   0 0 0
0 0 0   3 5 6   0 0 0      0 0 0   3 5 6   0 0 0      0 0 2   3 5 7   0 4 0
3 5 2   1 4 7   0 0 6      3 5 2   1 4 8   7 9 6      3 5 7   1 4 6   8 9 2

4 7 9   5 1 3   2 6 8      4 7 9   5 1 3   2 6 8      4 7 9   5 1 3   2 8 6
5 8 6   7 2 9   3 1 4      5 8 6   7 2 9   3 1 4      5 8 6   7 2 9   3 1 4
2 3 1   6 8 4   5 7 9      2 3 1   6 8 4   5 7 9      2 3 1   6 8 4   5 7 9


0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      9 4 3   8 6 5   7 2 1
0 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0      8 6 7   4 2 1   9 5 3
1 2 5   4 0 8   6 0 0      1 2 5   0 0 0   6 0 0      1 2 5   9 7 3   6 4 8

6 1 4   2 9 X   0 0 0      0 0 0   2 9 0   0 0 0      6 1 8   2 9 7   4 3 5
8 9 2   3 5 7   1 4 6      0 0 2   3 5 0   0 0 0      7 9 4   3 5 8   1 6 2
3 5 7   1 4 6   9 8 2      3 5 7   1 4 8   0 0 0      3 5 2   1 4 6   8 9 7

4 7 9   5 1 3   2 8 6      4 7 9   5 1 3   2 8 6      4 7 9   5 1 2   3 8 6
5 8 6   7 2 9   3 1 4      5 8 6   7 2 9   3 1 4      5 8 6   7 3 9   2 1 4
2 3 1   6 8 4   5 7 9      2 3 1   6 8 4   5 7 9      2 3 1   6 8 4   5 7 9
```

Sí que dedicarem més atenció a la 6ª representació, ni que sigui per la inquietant aparició de quatre marques X a la 7ª fila: ho hem fet per no complicar la notació amb un altre caràcter i és una manera d'indicar que també s'ha produït un atzucac, però no perquè de sobte quatre caselles s'hagin quedat sense candidats sinó perquè cap casella d'aquesta fila no té el valor 9 assignat ni el té com a candidat (X). A propòsit d'això, convé recordar que ja en el subcapítol *Etapa prèvia: crear una solució, II* havíem enunciat la **norma 1**, que s'havia de complir en files, columnes i requadres 3×3. El control anava a càrrec **d'EXPLORA-3*9**, que activava **FORA** (**POST**, n'havíem dit inicialment) quan s'infringia aquesta norma, i en *Etapa prèvia: crear una solució, III*, s'havia decidit que n'hi havia prou a aplicar-lo als requadres 3×3. En aquest mateix subcapítol havíem incorporat a **RE-MEMBER** un altre control: també calia activar **FORA** quan alguna casella es quedés sense candidats. Finalment, a *Etapa prèvia: crear una solució, IV* havíem recuperat el primer control per a les files i columnes, i tot plegat ho havíem reunit en **ACTUALITZA-PLUS**, on la detecció de caselles lliures sense candidats quedava a la vista en el cos d'aquesta funció, mentre que la de possibles infraccions a la **norma 1** l'encabíem en la subordinada **EXPLORA-9+9+5**, reclamada per files, columnes i requadres 3×3 amb un codi semblant:

```
  ..........
  (setq K 0)
  ..............................
  (mapcar '(lambda (E-9*9 E-LL3)
             (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
             ....................
          F-9*9 F-LL3)
  (if (= K 1)
      (PLUS-N->P J)
      (if (= K 0) (setq FORA T)))...
```

Immediatament ens assalta el dubte següent: ¿no hauríem de considerar també aquest tipus d'atzucac, als efectes del **CANVI**? Si hem de ser ser sincers, cal reconèixer que la incidència podria ocasionar una encallada tan aparatosa com la de qualsevol "atzucac puntual", però és a l'hora de definir el canvi en l'ordre de l'exploració de les caselles lliures que no sabríem com lligar-ho tot: sense anar més lluny, en el cas que ens ocupa, ¿quina de les quatre caselles marcades amb la X seria **NOLLL**?

Sortosament (?), si 7,4 (8) era el causant oficiós d'aquest atzucac no reconegut, a la 7ª representació el següent i últim candidat de la mateixa casella, 7,4 (9), sí que provoca un atzucac en la posició 6,6 que el converteix en **CAUSANT** oficial i, en retrocedir a 6,4 (8, a la 8ª representació), activa el **CANVI**. ¿Què hi fa, doncs, que aquest canvi no elimini la demora del veredicte, com passava fins ara?

0 0 0	0 0 0	0 0 0		0 0 0	0 0 0	0 0 0		0 0 0	0 0 0	0 0 0	
0 0 0	0 0 0	0 0 0		0 0 0	0 0 0	0 0 0		0 0 0	0 0 0	0 0 0	
1 2 5	0 0 X	0 0 0		**1 2 5**	0 0 X	0 0 0		**1 2 5**	0 0 X	0 0 0	
5 0 0	**2** 9 8	0 0 0		7 1 4	**2** 9 8	6 0 0		0 0 0	**2** 9 8	0 0 0	
8 9 0	**3 5** 6	0 0 0		6 9 8	**3 5** 7	0 2 0		9 8 0	**3 5** 7	0 0 0	
3 6 2	**1 4** 7	8 9 5		3 5 2	**1 4** 6	7 9 8		3 5 2	**1 4** 6	7 9 8	
7 8 9	5 1 3	2 4 6		5 7 9	8 1 3	2 4 6		5 7 9	8 1 3	2 4 6	
4 5 6	8 2 9	3 **1** 7		4 **8** 6	5 2 9	3 **1** 7		4 **6** 8	5 2 9	3 **1** 7	
2 3 1	6 7 **4**	**5** 8 9		**2 3 1**	6 7 **4**	**5** 8 9		**2 3 1**	6 7 **4**	**5** 8 9	

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	0 0 0
1 2 5	4 0 8	**6** 0 0
6 1 4	**2** 9 X	0 0 0
8 9 2	**3 5** 7	1 4 6
3 5 7	**1 4** 6	9 8 2
4 7 9	5 **1** 3	2 8 6
5 **8 6**	**7** 2 9	3 **1** 4
2 3 1	6 8 **4**	**5** 7 9

Doncs que, si ens fixem en l'atzucac que activava **CANVI** en els tres casos precedents, repetits just damunt d'aquestes línies, veureu que en reiniciar l'exploració començant per la casella marcada amb la X no fèiem altra cosa que influir directament sobre les candidatures d'una de les dues caselles ocupades de manera diversa a la solució NATURMÍN compatible: en tots tres casos hi havia 3 a 5,3 i 1 a 6,3, i el primer candidat a 6,7 (3) provocava un atzucac a 6,3 en emplenar-se 5,3 (1), cosa que obligava a passar immediatament a 5,3 (2) i a 6,3 (3); a partir d'aquí, ja en sintonia amb NATURMÍN compatible, l'emplenament no oferia cap problema.

Per contra, el **CANVI** resulta del tot inoperant en el cas que ens ocupa, l'atzucac del qual hem repetit a sota dels esmentats, a l'esquerra (llàstima que hagi quedat a la pàgina precedent!): reiniciar l'exploració a 6,6 no conduirà a res, perquè la diferència amb NATURMÍN compatible responsable de l'accidentada regressió des de 6,4 és el parell de caselles 5,2 (2, que ha de ser 3) i 7,2 (3, que ha de ser 2); però la casella 6,2 (9) que s'hi interposa, única influïda directament pels valors que se succeiran a 6,6, ja té el valor que ha de tenir. ¿Hi ha algun procediment senzill que no sols doni prioritat a l'exploració de la casella-atzucac sinó a la de les caselles lliures veïnes en la fila (posats a demanar, a totes les caselles lliures de la fila), assegurant una ràpida reordenació de les caselles d'acord amb la solució final? Doncs sí, i és tan senzill (i sona tant *a déja vu*) que fa riure: permutem els requadres 9×3 de manera que el que contingui la casella-atzucac quedi en posició inferior. Un corol·lari d'això és que els atzucacs que s'esdevinguin en el requadre 9×3 inferior mai no podran ser conflictius, criteri que ens permetrà estrènyer una mica més la condició per fixar provisionalment els punts **CAUSANT** i **NOLLL**, en **ACTUALITZA-PLUS**:

```
            (if (and PRIMER (not INI) (/ (cadr E-9*9) 0)
                     (setq I (car CAMI))
                     (not (or (= (cadr E-9*9) (cadr I))
                              (and (= (/ (car E-9*9) 3) (/ (car I) 3))
                                   (= (/ (cadr E-9*9) 3) (/ (cadr I) 3))))))
                (setq CAUSANT I NOLLL E-9*9))
```

Amb aquesta incorporació, un podria preguntar-se si no estan de més les condicions d'exclusió quan l'atzucac es troba a la mateixa fila o requadre 3×3 que la casella causant, atès que el cas que ens havia fet adonar de la necessitat d'apartar-los, per poder arribar a actuar sobre algun atzucac realment problemàtic, es produïa en el requadre 9×3 inferior. Però el que passava en aquest requadre pot passar en els altres dos (tot i que, quan més amunt, més probable serà que abans es produeixi un atzucac susceptible d'activar **CANVI**) raó per la qual caldrà mantenir la prevenció.

Donem per tancada aquesta qüestió menor i anem a **RE-MEMBER** i **MASCARA**, que són les funcions més afectades pel cop de timó. La primera funció s'ha simplificat, perquè ara **PRIMER** (a més de **INI**) només serveix per diferenciar els dos accessos, activada en el primer i desactivada (si s'escau, per haver-se activat **CANVI**) en el segon, i ja no hi ha excepció en l'ordre d'exploració dels escaquers 9×9, perquè la casella següent sempre és la primera lliure, localitzada per **COMPLET-9*9**. Diem escaquers, en plural, perquè **CANVI** haurà permutat els requadres 9×3, deixant el que contenia l'atzucac en posició inferior i mantenint la posició relativa dels altres dos:

```
(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P FIL COL)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              ..................................................
              (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P))))))
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (if PRIMER (setq CAMI (cons R-P CAMI)))
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (and (not OK) (not CANVI) (member E R-N))
              (progn
                (if PRIMER
                    (progn
                      (setq CAMI (member R-P CAMI))
                      (if (and NOLLL (not (member CAUSANT CAMI)))
                          (setq CANVI T))))
                (if (not CANVI)
                    (progn
                      (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                      (if (not FORA)
                          (if (equal (cadr 2R) NIL*NIL)
                              (setq OK T)
                              (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)
```

És a **MASCARA** on tindrà lloc la permutació de **RR-9*9** i **RR-LLL**, que farem aprofitant *mutatis mutandis* el dispositiu (basat en la variable **JJ** i la funció **F=3**) amb què, a la primera part d'aquesta funció i després de transposar ****9*9**** i **LLL** si calia, permutàvem els mateixos requadres 9×3 i les línies de cada requadre:

```
  ..............
  (foreach N L1A9
    (if POST (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                      "el nombre...\""))
    (if (and (member N L)
             (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                 (progn
                   (setq RE-MEM (RE-MEMBER 0-**9*9** 0-LLL))
                   (if CANVI
                       (progn
                         (setq JJ (if (= (/ (cadr NOLLL) 3) 1) '(1 0 2) '(2 0 1))
                               9**9 () I -1)
                         (foreach EE (F=3 RR-9*9 T)
                           (setq 0-L () I (1+ I))
                           (foreach E EE
                             (setq 0-L (cons (if (atom E) E (list (car E) I))
                                             0-L)))
                           (setq 9**9 (cons (reverse 0-L) 9**9)))
                         (setq RR-9*9 (reverse 9**9) RR-LLL (F=3 RR-LLL T)
                               PRIMER () CANVI ()
                               RE-MEM (RE-MEMBER RR-9*9 RR-LLL))))
                   RE-MEM)))
        (progn ...)))...
```

I, ja que hem fet esment de la transposició i les permutacions que ens preníem la molèstia de realitzar abans que **RE-MEMBER** ens facilités la relació dels candidats admissibles a la casella activada, reordenant l'escaquer en funció de la densitat d'ocupació precisament per optimitzar l'exploració, ara tocaria preguntar-nos si aquella feina no era inútil, atès que la penúltima modificació consistia en una transposició de **R-9*9** i **R-LLL** (dins de **RE-MEMBER**, convertides en **RR-9*9** i **RR-LLL**), si s'esdevenia que **(> COL FIL)**, i l'última consisteix a permutar requadres 9×3 si s'activa **CANVI** (per comoditat, aquesta operació s'efectua fora de **RE-MEMBER**, però no costaria gens muntar-la en aquesta funció, perquè com l'anterior limita el seu abast a cadascun dels valors candidats). La resposta gairebé ha quedat implícita en les precisions entre parèntesi: en la reordenació prèvia (pròpia de cada crida a **MASCARA**) només tenim en compte la distribució de caselles ocupades, incloent-hi la que anem a explorar; les que es practiquen després (pròpies de cada intervenció de **RE-MEMBER**, considerant la interrompuda i la represa com una mateixa execució), estan associades a cadascun dels candidats compatibles en primera instància i van a optimitzar l'exploració que s'inicia amb l'assignació virtual d'aquests valors **N** (**N** i la tapada automàtica de forats que pugui desencadenar en l'etapa **INI** = **T**, pel que fa a la transposició, i la primera embranzida en l'etapa **INI** = **nil**, pel que fa al desplaçament dels requadres 9×3). Així que, amb independència de l'efectivitat real de la reordenació prèvia (que algú amb interès pel tema podria escollir com a objecte de recerca), conceptualment no és contradictòria ni redundant en relació a les que segueixen: podríem parlar d'un preajust seguit d'ajustos més fins adequats a cada valor candidat.

Tanmateix, sí que hauríem de revisar una decisió presa no fa gaire i que ara, a la vista del nou rumb, ha quedat mancada de justificació. Ens referim a l'atzucac que dues pàgines enrera no savíem com tractar: una infracció contra la **norma 1** que es produïa a la 7ª fila, deguda a la no admissió del candidat 9 en la 5ª, 6ª, 8ª i 9ª casella, úniques desocupades. La perplexitat provenia de no saber com administrar a una fila-atzucac un remei preparat a la mida de caselles-atzucac, però ara que en comptes de donar prioritat a una posició li donem a tot un requadre 9×3, podem integrar aquests supòsits perfectament: no només podem sinó que caldria fer-ho, ja que ningú no ens garanteix que, com passava allà, immediatament després d'atemptar contra la **norma 1** sempre s'esdevindrà una casella-atzucac que salvarà la situació. Com veureu, l'adaptació del codi és ben senzilla, i l'única decisió addicional que haurem de prendre és restringir el tractament als casos d'infracció de la **norma 1** en files i requadres 3×3... entrant en contradicció amb allò que predicàvem només cinc línies enrera, perquè quan la infracció es produís en columnes ens tornaríem a quedar sense saber què fer. L'únic que pot adduir l'autor és que, per més que s'hi ha esforçat, no ha aconseguit donar forma a cap supòsit en què un veredicte

retardat fos responsabilitat d'una infracció a la **norma 1**, i ja és molt que, per prudència, s'hagi avingut a considerar dos de cada tres casos. D'altra banda, la decisió almenys és coherent amb una altra que havíem pres en el sentit d'excloure aquelles caselles-atzuzac en què la casella causant se situés a la mateixa fila o requadre 3×3, sense haver-ho fet amb les columnes: així, com que la integració de les files i requadres 3×3 infractors de la **norma 1** en el dispositiu existent la farem representant aquesta unitat modular per una de les caselles implicades (per la primera, com veureu), també ens assegurem que infraccions que poden superar-se provant altres candidats en la casella causant o efectuant regressions de curta volada no siguin motors d'un **CANVI** que no conduiria a res i deixin el camí expedit a atzucacs posteriors de més entitat.

Per tal d'evitar innecessàries duplicacions del codi, crearem la funció **ATZUCAC**:

```
(defun ATZUCAC (E)
  (if (and PRIMER (not INI) E (> (/ (cadr E) 3) 0)
           (setq I (car CAMI))
           (not (or (= (cadr E) (cadr I))
                    (and (= (/ (car E) 3) (/ (car I) 3))
                         (= (/ (cadr E) 3) (/ (cadr I) 3))))))
      (setq CAUSANT I NOLLL E))
  (setq FORA T))
```

L'enigmàtica presència del argument **E** sol (per això l'hem subratllat) just abans de **(/ (cadr E) 3) 0)**, a la segona línia, s'explica per la necessitat d'accedir-hi des d'**EXPLORA-9+9+5**, per executar **(setq FORA T)**, tant si **F-9*9** i **F-LL3** corresponen a files (**FIL/COL** = **1**), a columnes (**FIL/COL** = **2**) o a requadres 3×3 (**FIL/COL** = **nil**), però evitant que en el segon cas entrin en joc les variables **CAUSANT** i **NOLLL**. Tot queda clar a la nova versió d'**EXPLORA-9+9+5**, on hem subratllat la part modificada:

```
(defun EXPLORA-9+9+5 (FIL/COL / NN MN NM F-9 F-L)
  (foreach Ñ L1A9 (if (not (member Ñ F-9*9)) (setq NN (cons Ñ NN))))
  (if (> (length NN) 0)
      (progn
        (setq NN (reverse NN) MN (last NN))
        (if (/= FIL/COL 2)
            (foreach E F-3*3 (if (and (not NM) (listp E)) (setq NM E))))
        (foreach Ñ NN
          (if (not PLUS)
              (progn
                (setq K 0)
                (if (and INI (= Ñ MN)) (setq F-9 () F-L ()))
                (mapcar '(lambda (E-9*9 E-LL3)
                           (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
                           (if (and INI E-LL3 (= Ñ MN))
                               (progn
                                 (setq I ())
                                 (foreach M E-LL3 (if M (setq I (cons M I))))
                                 (setq F-9 (cons E-9*9 F-9) F-L (cons I F-L)))))
                        F-9*9 F-LL3)
                (if (= K 1)
                    (PLUS-N->P J)
                    (if (= K 0)
                        (ATZUCAC NM)
                        (if (and INI (= Ñ MN) (> (length F-L) 3))
                            (EXPL-SUBCONJ)))))))))
```

I, pel que fa al fragment d'**ACTUALITZA-PLUS** que havia quedat amb l'aspecte següent

```
  .................
  (setq K 0)
  (foreach E E-LLL
    (if (and (< K 2) E) (setq Ñ E K (1+ K))))
  (if (= K 1)
      (PLUS-N->P E-9*9)
      (if (= K 0)
          (progn
            (if (and PRIMER (not INI) (/ (cadr E-9*9) 0)
                     (setq I (car CAMI))
```

```
                    (not (or (= (cadr E-9*9) (cadr I))
                             (and (= (/ (car E-9*9) 3) (/ (car I) 3))
                                  (= (/ (cadr E-9*9) 3) (/ (cadr I) 3))))))
                (setq CAUSANT I NOLLL E-9*9))
            (setq FORA T))))...
```

quedarà reduït a

```
  ................
  (setq K 0)
  (foreach E E-LLL
    (if (and (< K 2) E) (setq Ñ E K (1+ K))))
  (if (= K 1)
      (PLUS-N->P E-9*9)
      (if (= K 0) (ATZUCAC E-9*9)))...
```

Donaríem per tancat el tema si no fos perquè les reiterades revisions de **RE-MEMBER** i de **MASCARA** ens han fet adonar d'un circumloqui absurd, que es va colar d'entrada en ambdues funcions i que fins ara havia passat inadvertit, perpetrat a l'hora de instrumentar el bucle en què només havien de ser processats els candidats 1... 9 admissibles en primera instància. A **RE-MEMBER**, l'expressió redundant

```
  ................
  (foreach E L1A9
    (if (and (not OK) (not CANVI) (member E R-N)) ...) ...)...
```

serà substituïda per

```
  ................
  (foreach E R-N
    (if (and E (not OK) (not CANVI)) ...) ...)...
```

I, a **MASCARA**, l'expressió

```
  ................
  (foreach N L1A9
    (if POST (RETOL-2 ...))
    (if (and (member N L)
             (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                 (progn ... RE-MEM)))
        (progn ...)))...
```

serà substituïda per

```
  ................
  (foreach N L
    (if POST (RETOL-2 ...))
    (if (and N (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                   (progn ... RE-MEM)))
        (progn ...)))...
```

I, posats a escombrar inconsistències que probablement havien tingut justificació en algun moment i que posteriorment van romandre per pura incúria i perquè tampoc no provocaven fallades, caldria traslladar les variables **C->F**, **INV-F**, **JJ**, **JJ1**, **JJ2** i **N/R-L** (però no pas **TN/R-L**) des d'**OMPLE-SUDOKU** a **MASCARA**. Amb això, la capçalera d'aquestes funcions quedarà així:

```
(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP*)

(defun OMPLE-SUDOKU (/ LLISTES 0*0 NIL*NIL CURSOR P-CURS TT TN/R-L)

********************************************************************************
********************************************************************************
********************************************************************************
```

I ara pareu atenció: hem dibuixat tres línies d'asteriscs per delimitar clarament el codi AutoLISP (que com sempre va en negreta) del present text explicatiu (que també escrivim en negreta però només per a una millor localització). Volem que a ningú que fullegi aquestes pàgines no li passi desapercebut el final de capítol (cloenda de 5 subcapítols que abasten més de la meitat del document, en pàgines), sobretot si té intenció de llegir els següents, perquè cal fer algunes precisions relatives a l'ordre de la seva elaboració i revisió, que havíem avançat en la 2ª pàgina de la *Introducció sense vaselina*: "de primer, una progressió que comença amb la relliscada 1, segueix des del 2 fins al 12 (saltant però, els 4, 5, 6 i 7), i s'amplia fins al 14; després un retorn al 3 i finalment la progressió definitiva fins al 14 (però incorporant-hi 4, 5, 6 i 7 i revisant tots els altres capítols)".

Ara que sabem com ha costat de preparar l'opció A d'obtenció d'una SUDOKU-SOLUCIÓ (improvisar-la sobre la marxa), molesta i menystinguda alternativa a les opcions B (copiar-la d'on fos) i C (partir de la CANÒNICA) que gairebé havíem estat temptats d'ignorar quan, un cop descartada per inviable l'orientació primitiva del treball, vam seguir creient força temps que l'única cosa a què valia la pena dedicar-se era a l'explotació d'alguna de les solucions (sense importar massa quina) per treure'n una munió de SUDOKUs-PROBLEMA, és l'hora d'aclarir el grau de desenvolupament en què es trobava aquesta opció ventafocs quan es va produir el retorn al capítol 3.

Vist com n'era de feixuc el primer mètode concebut per extreure un SUDOKU-PROBLEMA d'una SUDOKU-SOLUCIÓ (mètode que acabaria sent el 2) a *... però no hi eren totes les que ho són*, i descobert un camí alternatiu (que fou 1 per la seva indiscutible superioritat) i fins i tot una via ràpida per obtenir el SUDOKU-PROBLEMA tot just assolida una SUDOKU-SOLUCIÓ amb l'opció A (l'anomenarem oficiosament "mètode 0") a *Un nou camí i també una drecera*, les fallades i mancances d'aquesta opció es van posar en evidència i va fer-se imperatiu tornar a *Etapa prèvia: crear una solució*, que en realitat era un dossier encara buit pel que fa a text explicatiu (al menys *Condicions mínimes*, *... però no hi eren totes les que ho són* i l'esmentada *Un nou camí i també una drecera* estaven redactades a mitges, perquè un cop a punt el codi executable l'autor havia sentit la necessitat d'anar fixant idees), per revisar la feina feta (mentre el capítol es convertia en el subcapítol inicial *Etapa prèvia: crear una solució, I*), corregir-la i augmentar-la (subcapítols *II*, *III*, *IV* i *V*). Si fa no fa, podem afirmar que la versió proposada al final de *Etapa prèvia: crear una solució, I (embolica, que fa fort!)* és coetània del discurs que fa referència als mètodes 1 i 2 en el capítol clau *Un nou camí i també una drecera*, discurs que s'articula sobre FES-SUDOKU i les funcions derivades FES-SUDOKU-1 i FES-SUDOKU-2 (a *Solucions compatibles ho són totes les que hi són...* i a *... Però no hi eren totes les que ho són*, l'última encara du el nom de FES-PUBLIC), és a dir que més o menys reflecteix la situació en què es trobava la fase de presolució del programa (opcions A, B i C, les dues primeres a càrrec de FES-SOLUCIÓ, fins ara anomenada OMPLE-SUDOKU) just abans de la revisió. La consegüent adaptació de la segona part a la revisió de la primera no afectarà el que portem escrit (hauria quedat millor, però a hores d'ara a l'autor li falta motivació), sinó que quedarà com a afegitó (podeu prendre-vos-ho filosòficament, pensant que això no és un llibre seriós sinó un quadern de bitàcola), concentrat a més en el capítol en què es va produir la interrupció, *Un nou camí i també una drecera*, on procedirem de la manera següent:

- **Respectarem el text expositiu previ al canvi de timó, i l'il·lustrarem amb una versió íntegra del codi del programa, opció més oportuna que no sembla perquè almenys estàvem obligats a mostrar les funcions afectades pel canvi: algunes de FES-SOLUCIO, que cal fer accessibles des de FES-SUDOKU, i algunes de FES-SUDOKU, que cal adaptar a les adaptacions de FES-SOLUCIO, ocasió adient per recapitular.**

- **Tot seguit, i en relació a la l'última versió de FES-SOLUCIO (OMPLE-SUDOKU) del capítol *Etapa prèvia: crear una solució, III (roda el món i torna al Born)*, ens limitarem a mostrar el codi de les funcions afectades (aquelles de la primera part que han d'incloure expressions que les facin accessibles des de la segona, i les de la segona que calgui adaptar a les modificacions fetes en la primera), previ un text explicatiu que, com el que ara llegiu, anirà en negreta i estarà precedit per la triple línia d'asteriscs. No ens molestarem a fer el mateix en relació a *Etapa prèvia: crear una solució, II (la difícil simplicitat)*, perquè acabava d'una forma un pèl confusa, sense cap resultat concloent: de fet, *III* es va escindir de *II* artificiosament, per evitar que els subcapítols agafessin massa embalum, tot i que més tard, com que la bola de neu no parava de créixer, es va abandonar el criteri homogeneïtzador, donant lloc a les baluernes *IV* i *V*).**

- **Tot seguit, i en relació a la l'última versió de FES-SOLUCIO del capítol *Etapa prèvia: crear una solució, IV (... o a Camprodon)*, farem el mateix.**

- **Tot seguit, i en relació a l'última versió de FES-SOLUCIO del present capítol, *Etapa prèvia: crear una solució, V (la Seca, la Meca i la vall d'Andorra)*, farem el mateix.**

(La forma més senzilla d'identificar les tres versions prèvies a la definitiva és copsar l'abast diferent de SUD-MIN, funció objecte del capítol *Condicions mínimes*: en la 1ª intervé a les fases de presolució -opció A- i postsolució -mètode 1 i 2-, en la 2ª només a la de postsolució -mètode 1 i 2- i en la 3ª sols en el mètode 2.)

Aquesta còmoda manera d'actualitzar capítols (més còmoda per a l'autor que per al lector, és clar), a base d'afegir-hi pegats al final, s'ha acabat adoptant també en els dos últims. En *Assegurar-se una visualització correcta* convenia justificar el perquè finalment s'havia decidit adaptar el codi a ACAD 2011, quan la intenció manifestada inicialment era mantenir-lo adaptat a ACAD 2004 (amb ajustos puntuals per fer-lo també accessible des d'ACAD 2000). I a *Consells pràctics per evitar una executio precox* hem anat una mica més lluny: deixant-lo tal com havia quedat abans d'emprendre la revisió que ens va obligar a tornar al capítol 3, com a testimoni inacabat d'un propòsit en què no vam haver de perseverar perquè mentrestant havia sorgit una idea millor, impulsora d'una nova revisió (centrada aquesta segona en la fase de postsolució del programa), l'annex s'obre amb l'habitual triple línia d'asteriscs seguida d'un breu advertiment en negreta que ens informa precisament sobre el fet que el text subsegüent no manté el format diferenciador en negreta, que es reserva per a les expresions en AutoLISP, recuperant la notació de sempre. Aquesta transgressió de la norma establerta per a les actualitzacions a la primera revisió es justifica per la naturalesa més complexa del text expositiu, que ja no és l'adaptació de continguts previs a una revisió explicada en un altre lloc sinó la descripció d'una nova revisió, que acaba de prendre forma en el següent apartat *Epileg: això acaba igual que el conte de la sopa de còdols?*, no previst d'entrada.

Detectar solucions compatibles

Abans de presentar el codi corresponent, acabàvem el capítol 2 (*Tornem a començar: de la solució al problema*) avançant que, un cop completada l'accidentada gestació d'**OMPLE-SUDOKU** al llarg dels capítols 3 al 7, en el 8 abordaríem **FES-PÚBLIC**. L'ús de l'ambigu verb abordar era intencionat, perquè sabíem que l'estudi de la funció es prolongaria fins al capítol 10 (*Solucions compatibles ho són totes les que hi són...*) i que en l'11 (*... Però no hi eren totes les que ho són*) encara hauríem de superar un greu error de plantejament, però no volíem parlar de recorregut des del capítol 8 a l'11, per evitar que la fragmentació en capítols fos atribuïda a les mateixes raons que la d'**OMPLE-SUDOKU:** en aquell cas la dilació havia estat causada per un seguit d'insuficiències que s'anaven detectant pel camí i les corresponents esmenes, però en el de **FES-PÚBLIC** volíem dividir el procés en parts monogràfiques per a una millor comprensió (tret d'una revisió obligada, objecte del capítol 11).

Només començat el capítol 1 (p. 26), ja prefiguràvem en què consistiria el procés per obtenir SUDOKUs-PROBLEMA d'una SUDOKU-SOLUCIÓ, tot i que llavors ho fèiem des de la il·lusòria perspectiva de posseir la col·lecció completa d'aquestes últimes. Però, com que dues pàgines després apuntàvem dues estratègies diferents (les dues tenien en comú l'emplenament progressiu de l'escaquer 9×9 amb valors de la SUDOKU-SOLUCIÓ, cercant després de cada activació altres solucions compatibles), serà bo recordar-les ara, per tal de deixar clar quin dels dos camins seguirà **FES-PÚBLIC:**

- La segona (de fet, la primera a ser considerada i implementada, i que quan vam anomenar "mètode 1" una de més eficient que se'ns va ocórrer en el capítol 12, va passar a ser el "mètode 2") és la que descrivim en aquest capítol i els dos següents, esmenarem en el capítol 11 (fins llavors s'identifica amb **FES-PÚBLIC**, atés que no hi ha altres opcions) i que, a partir del 12 (en què aquesta funció passa a anomenar-se **FES-SUDOKU**) comparteix protagonisme amb mètodes alternatius. A diferència de l'altre, on ens conformarem a saber si hi ha una o més solucions compatibles amb la part ja activada, en aquest mètode ens vam empatollar a saber quantes n'hi havia i quines eren, sense haver-hi cap necessitat (fora d'un petit avantatge secundari: tenir l'usuari permanentment informat sobre el particular, cosa que, amb la possibilitat de desactivar qualsevol activació prèvia i seguir un camí diferent, pot constituir un ajut en la consecució dels seus propòsits). No cal insistir que aquest objectiu presenta molts obstacles, el principal dels quals és la ingent quantitat d'informació que haurem de remenar al començament: sense anar més lluny hem de pensar que, si amb totes les caselles desactivades el nombre de solucions compatibles és de **6.670.903.752.021.072.936.960**, perquè totes les SUDOKUs-SOLUCIÓ són compatibles, en iniciar el procés activant-ne una el nombre "es redueix" a la novena part, situació que segueix sent inabordable; res no hi fa que ens limitem a solucions normalitzades, perquè dividir aquestes xifres per **1.296** tampoc no soluciona cap problema. Això fa que en aquest mètode sigui imperativa la instauració d'un llindar d'ocupació (tema que ja hem tractat en els últims cinc capítols però en relació a l'opció A d'obtenció de la SUDOKU-SOLUCIÓ, i que en el capítol següent tornarem a tractar, aquesta vegada lligat a l'obtenció d'un SUDOKU-PROBLEMA a partir de l'anterior), per sota del qual podem activar caselles despreocupadament, en la certesa que no arribarem a cap SUDOKU-PROBLEMA; si no, l'exigència de remenar des del principi solucions compatibles hauria d'anar aparellada a la prèvia disponibilitat d'algun inventari (¿o és que pretenem construir sobre la marxa un inventari que sabem que pot trigar anys a completar-se?). La idea és que, en el moment de passar el llindar, el conjunt de solucions compatibles ja s'hagi reduït considerablement i sigui més abordable.

- La primera consisteix a escatir, després de cada activació, si només la nostra SUDOKU-SOLUCIÓ és compatible amb la part pública del sudoku (totes les caselles activades, incloent-hi l'última) o n'hi ha més (tenint en compte que, només que n'haguem trobat una altra, ja n'hi ha prou per afirmar-ho i continuar activant caselles); naturalment, el SUDOKU-PROBLEMA queda determinat amb aquella casella en què no n'hi hagi cap més. El gran avantatge d'aquest procediment és que, tot i que la dificultat de trobar solucions compatibles va augmentant amb el nombre de caselles activades (després d'activar la primera casella, de seguida trobarem una solució compatible diferent de l'escollida), perquè cada nova casella és un obstacle més en l'exploració (culminant en l'última, en què no podrem assegurar que la SUDOKU-SOLUCIÓ que hem adoptat sigui l'única compatible amb les caselles

activades mentre no haguem provat tots els valors possibles en les caselles que resten sense activar), també és cert que cada cop el territori que cal explorar és més reduït, circumstància que fins a cert punt contraresta l'anterior. A més, tot i no ser obligada com en l'estratègia precedent, la instauració d'un llindar contribuiria a alleugerir les primeres activacions. Dissortadament, una idea tan elemental com aquesta es va creuar diverses vegades en el camí de l'autor, sense que aquest li prestés massa atenció, en entendre erròniament que aquest sistema era una simple variant d'allò que el capítol 12 homologaria com a "mètode 1", i similar en eficiència. **Aquest comentari és l'única infracció de l'autor al pacte segellat en negreta al final de l'últim capítol (pàg. 355 a 357); caldrà esperar a la fi de la fi (final de l'*Epíleg: això acaba igual que el conte de la sopa de còdols?*) perquè recuperem i desenvolupem aquesta idea**. I ara, seguim com si res.

Centrats ja en la que hem presentat com a segon mètode, i d'acord amb allò que hem dit a l'inici del capítol, farem una primera aproximació a **FES-PUBLIC** per veure'n l'estructura general i estudiar la funció **CLIC** (que interpreta i elabora els clics efectuats en les caselles de l'escaquer 9×9 des de qualsevol d'ambdues finestres); deixarem per al capitol 9 la descripció de la funció **SUD-MIN** (que marca el llindar d'emplenament a partir del qual caldrà comptabilitzar les solucions compatibles) i completarem la crònica en el capítol 10, amb el tàndem **PERFIL-V*V** + **COMPAT-PERFILS** (funcions que, en rebre el senyal **POST** = **T** de **SUD-MIN**, s'ocuparan de fer el primer inventari de solucions normalitzades compatibles amb la part pública de la V. E. NORMALITZADA), i la funció solitària **NUMCOMPAT** (que anirà actualitzant a la baixa l'inventari, a mesura que activem noves caselles, fins que només quedi una solució compatible), tot i que caldrà esperar a desfer un equívoc en el capítol 11 perquè **FES-PUBLIC** adopti la forma definitiva (definitiva efímerament, perquè el 12 durà novetats, entre elles la incorporació de l'assenyalat com a primer mètode, moment en què **FES-PUBLIC** canvia a **FES-SUDOKU**, desdoblada en **FES-SUDOKU-1** i **FES-SUDOKU-2**).

Després de les inicialitzacions de ****V*V**** i ****V**V****, comentades en el capítol 2, canviarem d'ubicació els valors ****9*9**** i ****9**9****, visualitzats en les finestres V. E. NORMALITZADA (esquerra) i VARIANT ESCOLLIDA (dreta) perquè la funció **GRAF-9*9** els havia dibuixat en les capes NORM-1 i VAR-1, passant-los a les capes NORM-0 i VAR-0, respectivament. La conseqüència immediata del canvi és que aquests valors passaran de veure's en blanc a fer-ho en gris, i la missió de l'usuari consistirà a restaurar la blancor en certes caselles, fins que el conjunt de valors en blanc determini els valors en gris. Com sabeu, l'activació de caselles fa que en ****V*V**** i ****V**V**** les coordenades d'aquestes caselles siguin substituïdes pel seu valor, però el canvi d'aspecte ja no serà un simple retorn dels caràcters numèrics a les capes NORM-1 i VAR-1, sinó que el contingut de les capes NORM-0 i VAR-0 romandrà de manera permanent i els caràcters activats es copiaran en les altres dues capes. Com que creiem que les instruccions rebudes a **EXPLICACIONS** ja eren prou clares, no ens hem molestat a requerir des de l'àrea inferior de text que l'usuari faci clic. En lloc d'això, sobre la franja gris que enllaça les dues finestres hi apareixerà una icona que molts recordaran de la bandera de Corea del Sud i que és un símbol taoista que representa l'etern devenir, la dialèctica dels contraris, el *yin* i el *yang*, el dia i la nit o com més us quadri, que girarà sobre ell mateix quan mogueu el cursor a l'Espai Model (tant és que ho feu des de la finestra dreta o esquerra) per anar a la casella que voleu activar i la velocitat del qual podreu regular amb la variable **ANG**, a l'inici de **C:SUDOKULUM**. La idea és que, cada cop que aparegui aquest símbol, l'usuari s'adoni que **FES-PUBLIC** ja ha enllestit l'última operació i espera un nou clic. La funció **while** és el motor d'iteracions i la interrupció es produeix quan polsem la tecla <Intro>, < > o el botó dret de la rateta, entrades que es detectaran amb la funció **grread**. Si mentrestant movem el cursor, es posa a girar el disc, fabricat prèviament per la funció **YIN-YANG** (que també crea el bloc **"C"**, tapadora circular que oculta aquest disc cada vegada que fem clic, restaurant la plenitud de la franja gris que va de finestra a finestra), i en fer clic sobre alguna casella (amb el botó esquerre, és clar), la posició queda enregistrada a **P**, també gràcies a **grread**. **P** representa d'entrada el punt marcat, i **PX** i **PY** cadascuna de les coordenades, però finalment (i col·laborant-hi la funció **PUNT**) tindrem que:

- **P** representarà el centre de la casella.
- **X** i **Y** seran les coordenades de **P**.
- **PX** i **PY** seran les posicions inferior-esquerra i superior-dreta, respectivament, del quadrat amb centre a **P** i costat **2 × 0,48 = 0,96** de què parlàvem en el capítol 2 (p. 93), dintre del qual s'ha d'haver produït el clic.

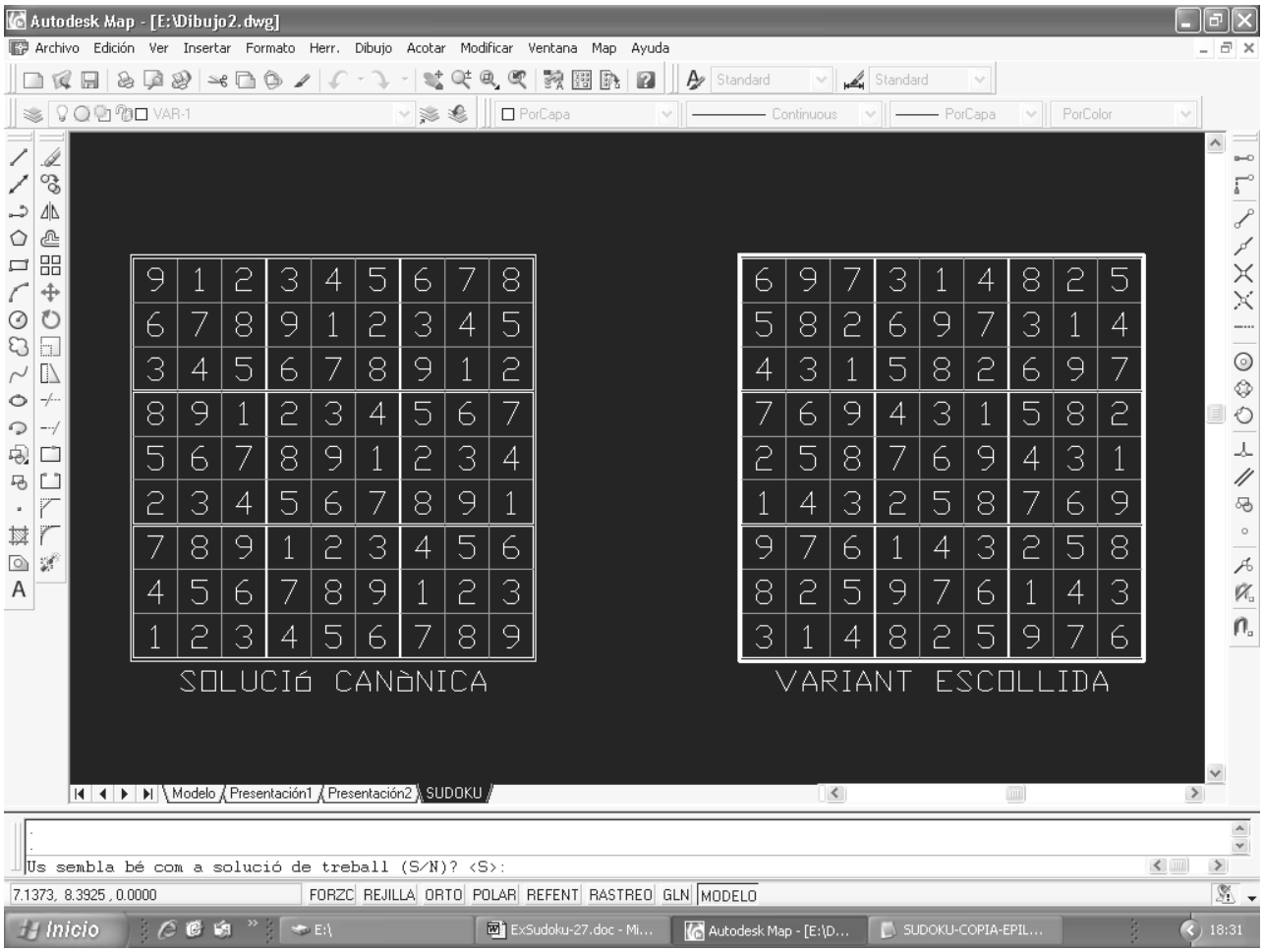
Amb aquests valors, i tenint en compte que venim d'una situació en què **POST** = **nil**, descriurem algunes de les incidències que poden esdevenir-se a cada iteració, fins que una resposta nul·la (fora de **SEGUIR** i de les previstes a **getkword**) hi posi fi.

Entre altres coses que veurem de seguida, la funció **CLIC** determinarà si la casella situada a **P** és visible o pública (**V** = **T**) i, per tant, en haver-hi fet clic a sobre volem desactivar-la, o estava desactivada (**V** = **nil**) i, per tant, volem activar-la. En començar (**(not POST)**), i abans de creuar el llindar, si és **(not V)** accedirem a **SUD-MIN** per veure si ja hi som. Quan per fi el creuem, **POST** s'activarà, esborrarem el símbol del Tao, inserirem la tapadora **"C"** sobre el forat que haurà deixat, per restablir la continuïtat de la franja gris, i sobre aquest fons veurem apareixer, esbalaïts, el rètol *Determinar quantes solucions normalitzades són compatibles pot trigar hores. Espereu...* Serà quan es posaran en marxa les funcions **PERFIL-V*V** i **COMPAT-PERFILS**, i haurem de suportar una llarga espera (que en versions posteriors ja no arribarà a durar hores, però mantindrem aquest text per curar-nos en salut), abans de saber quantes solucions normalitzades encara són compatibles amb la part pública, i seguir activant caselles. A partir d'aquest moment, cada nova activació serà analitzada per **NUMCOMPAT**, que ens anirà tenint al corrent de les solucions normalitzades encara compatibles (aquestes intervencions són molt més curtes i no necessiten cap advertiment), fins que ens arribi la bona nova *EL SUDOKU JA TÉ UNA SOLUCIÓ NORMALITZADA ÚNICA. Omplint més caselles el fareu més fàcil (<Intro> o < > per acabar)* i polsem alguna d'aquestes tecles per sortir de **while** i de **FES-PUBLIC**. Les complicacions, en cada una d'aquestes etapes, i les corresponents commutacions de valor de **POST** seran fruit de clics practicats en situació **V** (desactivacions) i les estudiarem dos capítols més endavant, perquè estan lligades a la problemàtica de **PERFIL-V*V** i **COMPAT-PERFILS**, de **NUMCOMPAT** i de les "fotos" de l'escaquer 9×9 fetes quan entra en acció **PERFIL-V*V** (en el moment de trepitjar el llindar) i quan **NUMCOMPAT** intervé per última vegada (en assolir-se la solució normalitzada única). Aquí ens limitarem a presentar el codi de **FES-PUBLIC** i funcions derivades, tret de **SUD-MIN** i de les tres que acabem d'esmentar, després d'explicar com funciona **CLIC**.

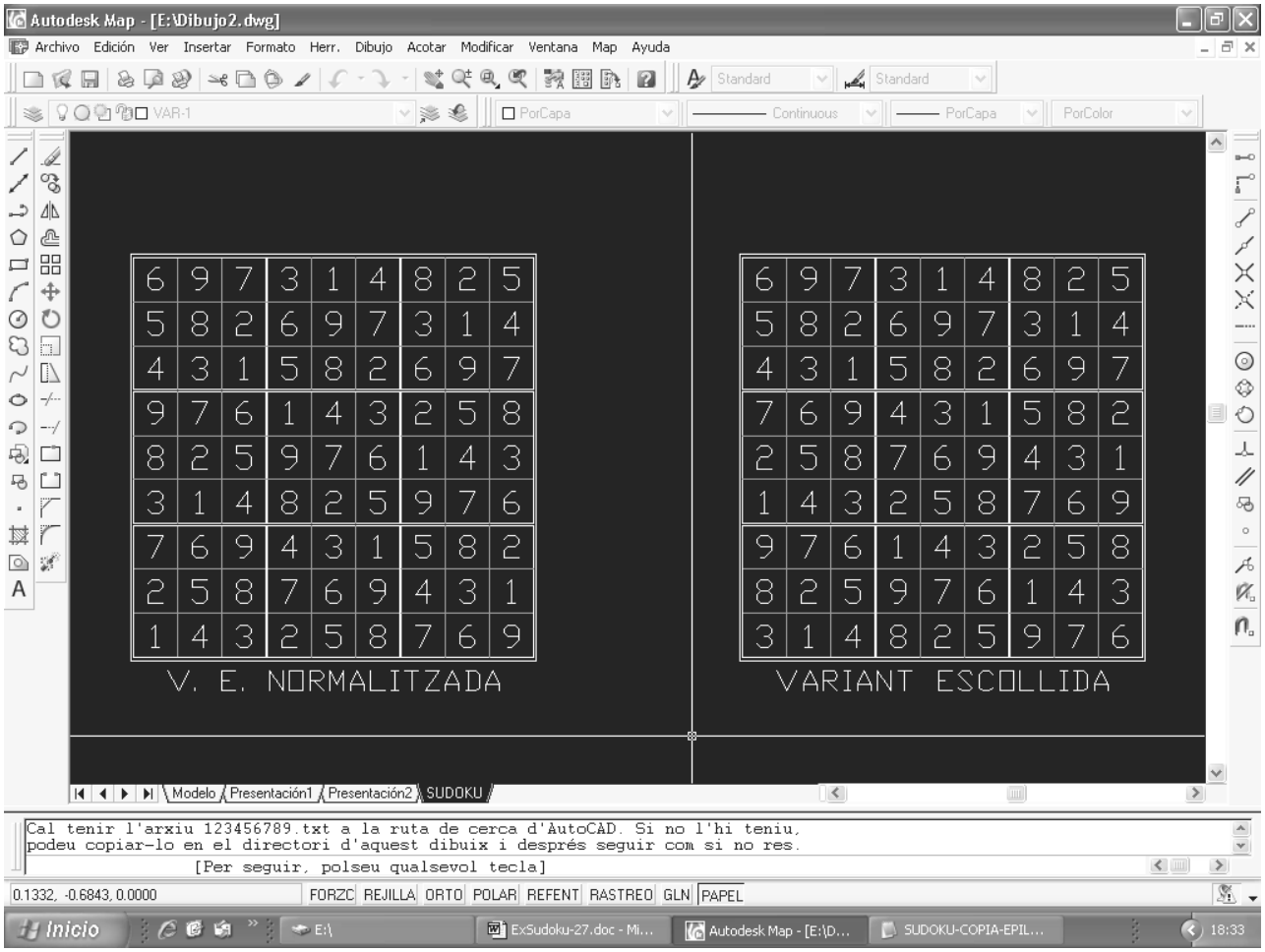
Ampliant l'esbós que havíem avançat en el capítol 2, **CLIC** utilitza les variables **P**, **X**, **Y**, **PX** i **PY**, presentades al final de l'última pàgina, per capturar de ****9*9**** o ****9**9**** el valor **N** de la casella on hem fet clic, deduir de ****V*V**** o ****V**V**** si estava activada i la desactivem (**V**) o a l'inrevés (**(not V)**), cosa que dependrà de si l'element corresponent a la casella és **N** o **(X Y)**, commutar aquest valor i reflectir gràficament aquesta commutació mitjançant la funció **V+-** que, si activem la casella en treu una còpia i la canvia de capa (de NORM-0 a NORM-1, o de VAR-0 a VAR-1) perquè veiem **N** en blanc, i si la desactivem esborra aquesta còpia perquè només veiem l'original en gris. Naturalment, de cada parell de pseudomatrius 9×9 o de capes esmentades, utilitzarem la que correspongui, segons que **(getvar "CVPORT")** valgui **2** o **3**, és a dir, segons que el clic l'haguem fet sobre la finestra esquerra (V. E. NORMALITZADA) o sobre la dreta (VARIANT ESCOLLIDA). Però l'acció (interna i externa) s'ha de transferir a l'altra finestra, i aquí és on haurem de recórrer a les funcions **N->V** i **V->N**, que utilitzen els paràmetres **N1**, **N2**, **N3** i **N4** obtinguts a **PRE-NORM->VAR** per obtenir la **Y** homòloga (el procés de normalització no afecta la **X** de les caselles) i repetir el procés a l'altra banda, prèvia la modificació de la variable de sistema **"CVPORT"** per canviar de finestra. Tot i que en el capítol 2 és on havíem tractat de **PERMUT-N** i **PRE-NORM->VAR**, presentant **N1**, **N2**, **N3** i **N4**, ara és el moment de comentar el paper que, mitjançant les funcions auxiliars **Y1**, **Y2** i **Y3**, hi juguen aquests paràmetres:

- A **N->V** (el clic s'ha efectuat en una casella de la V. E. NORMALITZADA, i la **Y** transformada correspon a la VARIANT ESCOLLIDA):
  - Amb la **Y** primitiva s'obté (mitjançant **Y1**, **Y2** o **Y3**, segons que aquesta pertoqui a la primera, segona o tercera fila del requadre 9×3) la **Y** de la fila on va a parar per efecte de la permutació interna del requadre, però sense considerar encara el desplaçament degut a la permutació de requadres 9×3.
  - Amb la nova **Y** s'obté (mitjançant **Y1**, **Y2** o **Y3**, segons que aquesta pertoqui al primer, segon o tercer requadre 9×3) la **Y** de la fila on va a parar finalment, considerant ara la permutació de requadres 9×3.

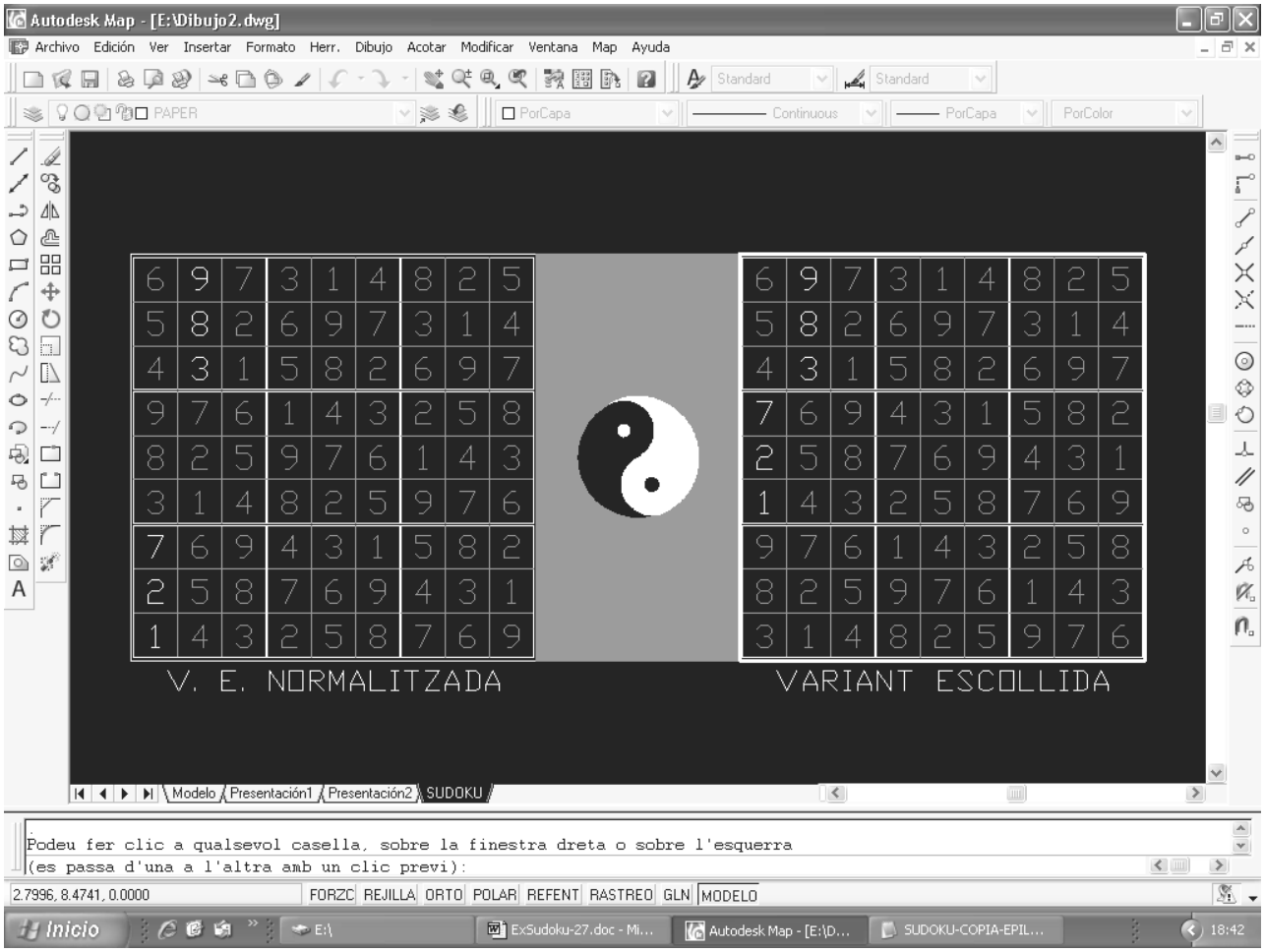
- A **V->N** (el clic s'ha efectuat en una casella de la VARIANT ESCOLLIDA, i la **Y** transformada correspon a la V. E. NORMALITZADA):
  - Amb la **Y** primitiva s'obté (mitjançant **Y1**, **Y2** o **Y3**, segons que aquesta pertoqui al primer, segon o tercer requadre 9×3) la **Y** de la fila on va a parar per efecte de la la permutació de requadres 9×3, però sense considerar encara la permutació de files en el interior del requadre.
  - Amb la nova **Y** s'obté (mitjançant **Y1**, **Y2** o **Y3**, segons que aquesta pertoqui a la primera, segona o tercera fila del requadre 9×3) la **Y** de la fila on va a parar finalment, considerant ara la permutació de requadres 9×3.



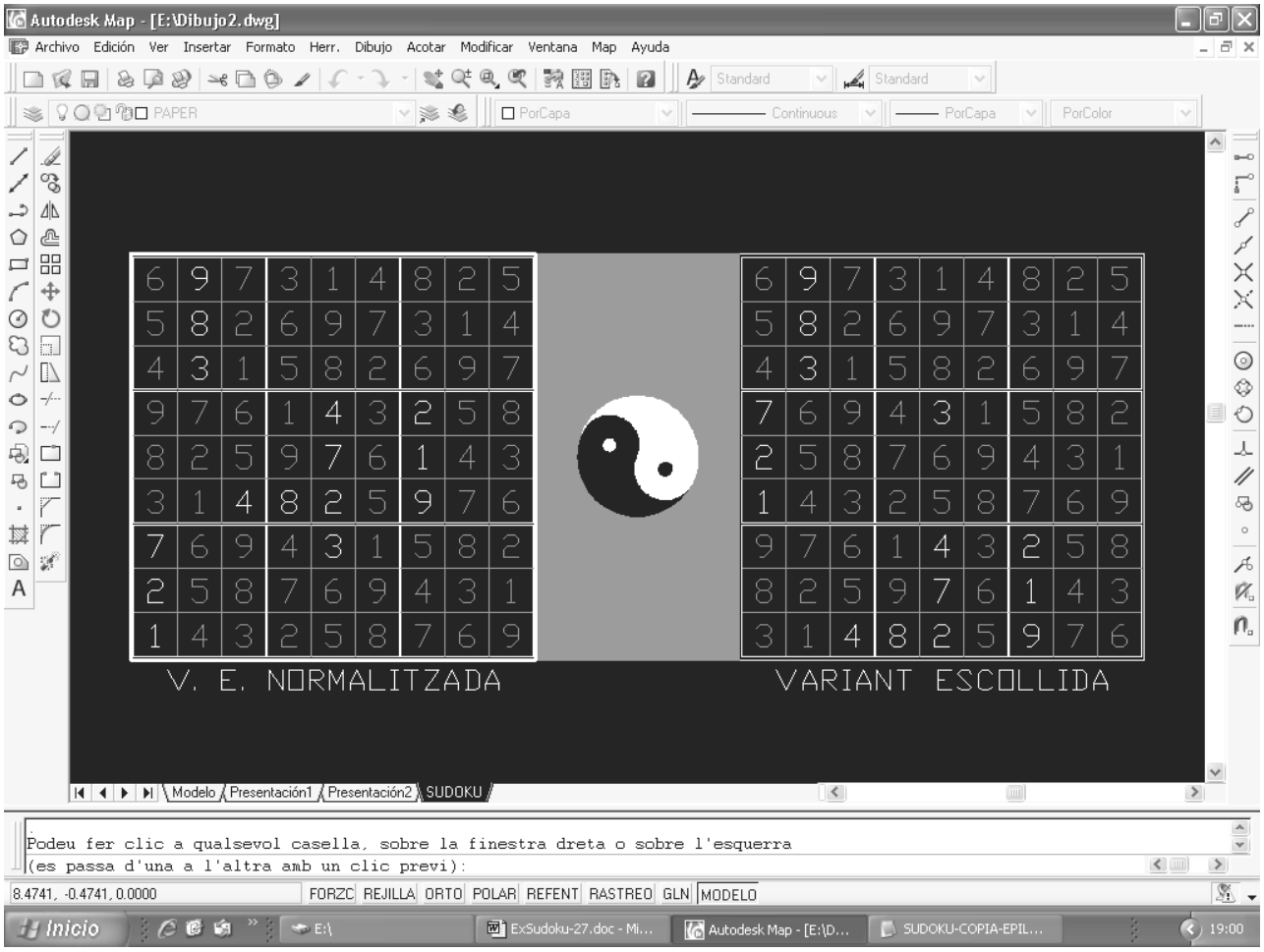
(ve de p. 90) Si la VARIANT ESCOLLIDA és acceptada, la finestra esquerra passa...



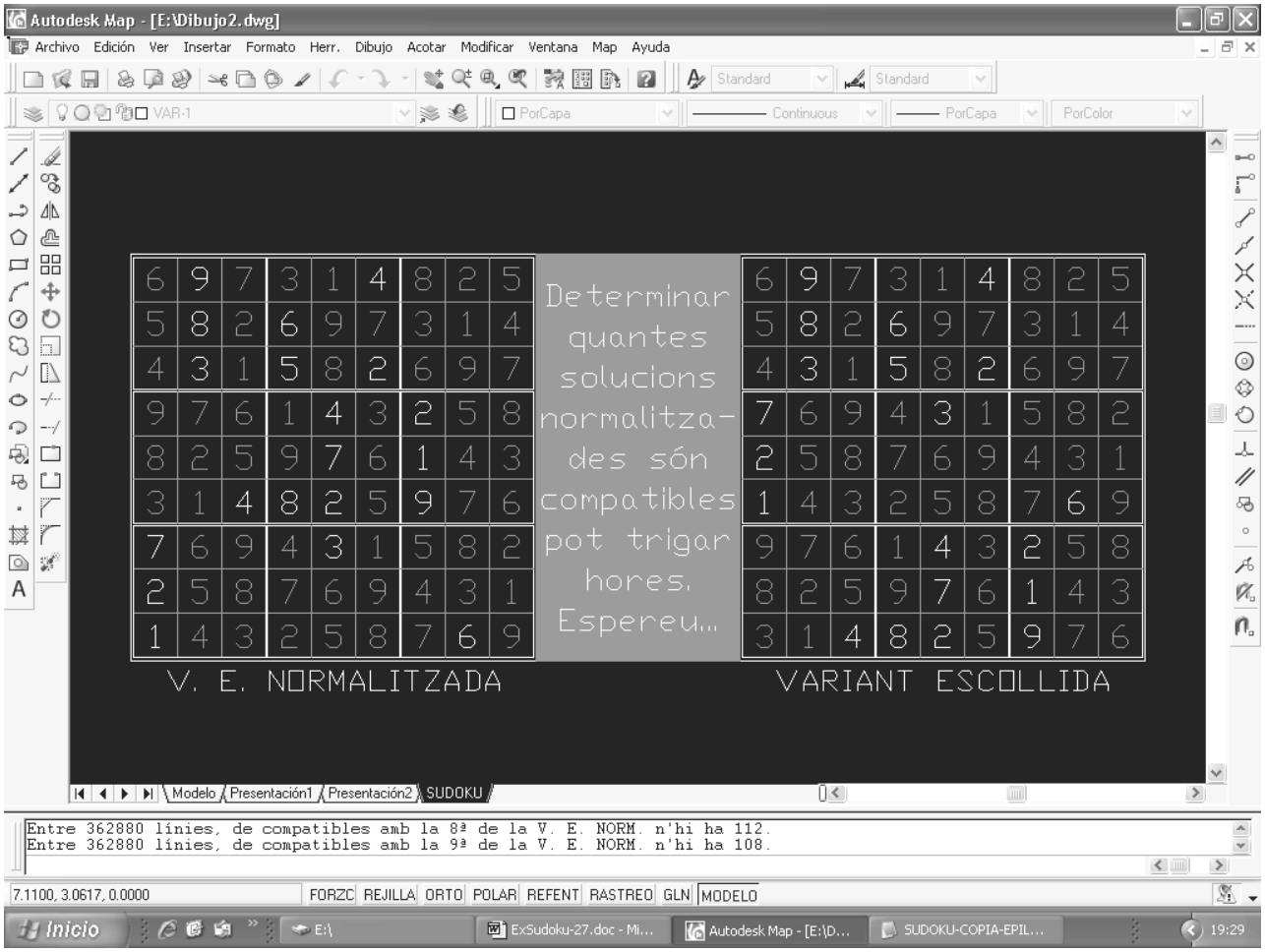
a ser ocupada per la versió normalitzada, sobre la qual es desenvolupa el procés.



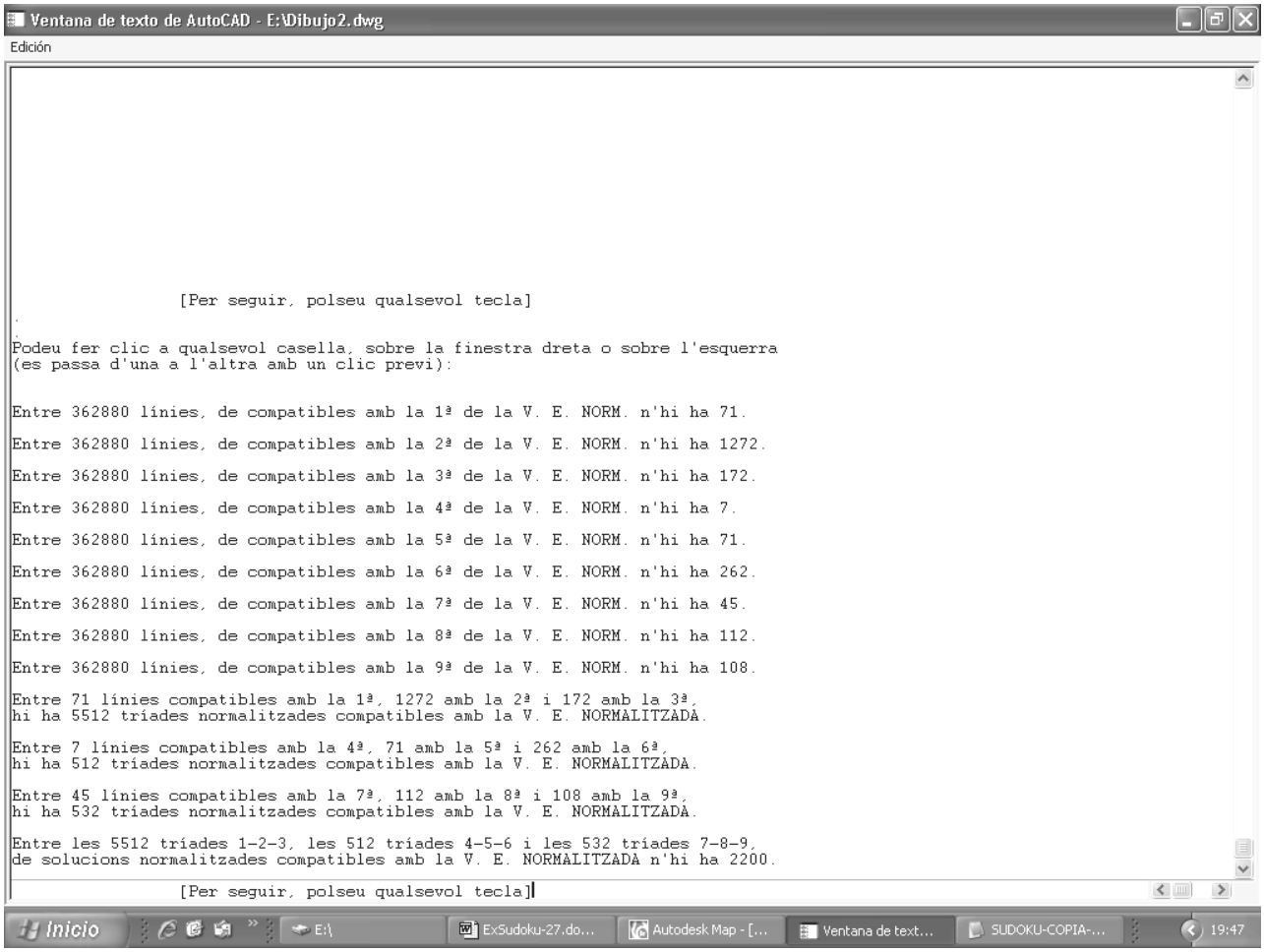
Malgrat això, es pot actuar en qualsevol finestra: les 6 primeres activacions...



s'han efectuat en la dreta, les 9 següents s'hsn efectuat en l'esquerra i les 5...



últimes s'han fet de nou en la dreta, moment en què es posa en marxa el còmput...



de solucions normalitzades compatibles, que acaba commutant a pantalla de text.

Com que el clic el podem efectuar des de qualsevol de les dues finestres (previ un clic addicional, si s'escau, per canviar de finestra de treball) convindrà aclarir uns dispositius que no s'expliquen amb la descripció succinta que hem fet de **CLIC**. Així, com que a la sortida de la funció **P**, **X**, **Y**, **PX** i **PY** han d'estar referits a la finestra esquerra (V. E. NORMALITZADA) per poder treballar amb ****V*V****, passa que:

- Quan fem clic a la finestra esquerra, la captura de **N**, determinar **V** (saber si la casella està activada i cal desactivar-la, o a l'inrevés), commutar el contingut de la casella en ****V*V**** (canviar **N** per coordenades o a l'inrevés) i formar el conjunt de selecció **SS** per anar a **V+-** i visualitzar aquests canvis, es fa primer en aquesta finestra. Després, mitjançant **N->V**, trobem els nous valors **P**, **PX** i **PY** corresponents a la finestra dreta (no cal repetir l'obtenció de **N** i **V**, que són iguals), commutem el contingut de la casella en ****V**V**** i formem **SS** per anar a **V+-** i visualitzar aquests canvis. Ara bé: els valors que surtin de **CLIC** han de ser els **P**, **X**, **Y**, **PX** i **PY** primitius, corresponents a la primera finestra; és per això (per poder-los recuperar al final) que, abans de **N->V**, els haurem assignat a sengles variables de transició **Q**, **QX** i **QY**.
- Quan fem clic a la finestra dreta, la captura de **N**, la determinació de **V** (saber si la casella està activada i cal desactivar-la, o a l'inrevés), la commutació del contingut de la casella a ****V**V**** (canviar **N** per coordenades o a l'inrevés) i formar el conjunt de selecció **SS** per anar a **V+-** i visualitzar aquests canvis, es fa primer en aquesta finestra. Després, mitjançant **V->N**, trobem els valors **P**, **PX** i **PY** corresponents a la finestra esquerra (no cal repetir l'obtenció de **N** i **V**, que són iguals), commutem el contingut de la casella de ****V*V**** i formem **SS** per anar a **V+-** i visualitzar aquests canvis. Com que aquests últims valors **P**, **X**, **Y**, **PX** i **PY** ja ens estan bé, no caldrà usar **Q**, **QX** i **QY**.
- Encara trobareu una altra cosa que us pot semblar arbitrària, i és que, per no entretenir-nos modificant **CLIC** gradualment, al llarg dels propers capítols, ja l'oferirem adaptada a la versió de **FES-PUBLIC** del capítol 11 (*...però no hi eren totes les que ho són*). Per bé que de moment només hem avançat que en situacions determinades treuríem "fotos" de l'escaquer 9×9 i, sent estrictes, és prematur referir-nos a fotos concretes, allà veureu que, si desactivem alguna casella que formi part de **FOTO1** o **FOTO2**, se'ns demanarà que confirmem una decisió que podria comportar una llarga espera. Si davant d'això decidim de fer-nos enrera, caldrà activar en ambdues finestres la casella desactivada, i ho aconseguirem repetint l'últim **CLIC**, però segons la finestra on l'haguem executat, això durà problemes:
  - Si ho havíem fet a la finestra esquerra, no hi haurà cap contratemps, perquè els valors **P**, **X**, **Y**, **PX** i **PY** subministrats al segon **CLIC** seran els que resultin del primer; de passada, haurem commutat el contingut de la casella ****V*V**** i haurem visualitzat el canvi.
  - Si ho havíem fet a la finestra dreta, sí que tindrem problemes, perquè hauríem de disposar dels valors **P**, **X**, **Y**, **PX** i **PY** corresponents a aquesta finestra. Per no haver de recórrer un altra vegada a **V->N**, una possibilitat seria que en el primer **CLIC** uséssim les variables de transició **Q**, **QX** i **QY**, malgrat no tenir-ne necessitat, per poder-les recuperar quan convingués. Però per fer això hauríem de tenir algun senyal que ens indiqués si es donava la circumstància descrita. Doncs bé, n'hi haurà prou amb el valor binari (**T** o **nil**) de **Q**: si a l'inici de cada iteració **while** fem **(setq Q ())**, i tenim la precaució de fer el mateix en acabar **CLIC**, quan la finestra de treball sigui l'esquerra, només podrà succeir **Q** = **T**, en començar una execució de **CLIC**, quan aquesta execució sigui la segona d'una mateixa iteració **while**, i la primera també hagi acabat amb aquest valor.

I, ja sense més preàmbul, us oferim el codi de les funcions que han anat sortint:

```
(defun YIN-YANG ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "ARCO" "0,-0.15" "0.15,0" "0,0.15" "ARCO" "" "0,-0.15"
           "ARCO" "0,0" "0.075,0.075" "0,0.15"
           "ARCO" "0,0" "-0.075,-0.075" "0,-0.15"
           "CIRCULO" "0,0.075" 0.015 "CIRCULO" "0,-0.075" 0.02
           "COLOR" G  "SOMBREA" "S" "-0.15,0" "0.15,0" ""
           "BLOQUE" "C" "0,0" "LT" ""
           "SOMBREA" "S" "0,0.5" "-0.15,0" "0.15,0" ""
           "BORRA" "LT" ""
           "COLOR" 7 "SOMBREA" "S" "0.15,0" "-0.075,-0.075" "0.075,0.075"
                                   "0,-0.055" "0,0.06" ""
```

```
           "UY" "DESIGNA" "P" "LT" ""
           "BORRA" "C" "-0.2,0.6" "0.2,-0.6" "E" "P" ""
           "DESIGNA" "LT" ""))

(defun COMMUTA (LL)
  (subst (subst (if V (list X Y) N) (if V N (list X Y)) (nth Y LL))
         (nth Y LL) LL))

(defun V+- (C)
  (if V
      (command "BORRA" (ssname SS 0) "")
      (command "COPIA" SS "" "0,0" ""
               "CAMBPROP" "LT" "" "C" C "")))

(defun Y1 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 0) N1) ((= Y 3) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (+ Y K)
              (if (or (= N K3) (= N K4))
                  (+ Y (* 2 K))
                  Y))))

(defun Y2 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 1) N1) ((= Y 4) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (- Y K)
              (if (or (= N K3) (= N K4))
                  (+ Y K)
                  Y))))

(defun Y3 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 2) N1) ((= Y 5) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (- Y K)
              (if (or (= N K3) (= N K4))
                  (- Y (* 2 K))
                  Y))))

(defun PUNT () (setq P (list X Y) PX (mapcar '- P TG) PY (mapcar '+ P TG)))

(defun V->N (/ N)
  (cond ((< Y 3) (Y1 3 2 3 4 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 4 1 3))
        (T (Y3 3 1 4 3 5)))
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 3 4 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 4 1 3))
        (T (Y3 1 1 4 3 5)))
  (PUNT))

(defun N->V (/ N)
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 4 3 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 3 1 4))
        (T (Y3 1 1 3 4 5)))
  (cond ((< Y 3) (Y1 3 2 4 3 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 3 1 4))
        (T (Y3 3 1 3 4 5)))
  (PUNT))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if Q (setq P Q X (car P) Y (cadr P) PX QX PY QY))
        (setq N (ELEMENT **9**9**)
              V (atom (ELEMENT **V**V**))
              **V**V** (COMMUTA **V**V**)
              SS (ssget "_C" PX PY))
        (if (not Q) (setq Q P QX PX QY PY))
        (V+- "VAR-1")
```

```
        (command "CVPORT" 2)
        (V->N)
        (setq **V*V** (COMMUTA **V*V**)
              SS (ssget "_C" PX PY))
        (V+- "NORM-1")
        (command "CVPORT" 3))
      (progn
        (setq N (ELEMENT **9*9**)
              V (atom (ELEMENT **V*V**))
              **V*V** (COMMUTA **V*V**)
              SS (ssget "_C" PX PY) Q P QX PX QY PY)
        (V+- "NORM-1")
        (command "CVPORT" 3)
        (N->V)
        (setq **V**V** (COMMUTA **V**V**)
              SS (ssget "_C" PX PY))
        (V+- "VAR-1")
        (command "CVPORT" 2)
        (setq P Q X (car P) Y (cadr P) PX QX PY QY Q ())))))

(defun FOTO-V*V (/ FOTO)
  (setq J -1)
  (foreach L **V*V**
    (setq I -1 J (1+ J))
    (foreach E L
      (setq I (1+ I))
      (if (atom E) (setq FOTO (cons (list I J) FOTO)))))
  FOTO)

(defun FES-PUBLIC (/ KK KKK Q QX QY V GR L1 SUDOKUS-V*V S-V*V FOTO1 FOTO2 *K*
                     *KK*)
  (setq **V*V** (INI-COORDS) **V**V** **V*V**)
  (prompt "\n.\n.\n.")
  (graphscr)
  (command "CVPORT" 2 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""
           "CVPORT" 3 "CAPA" "D" "NORM-1" ""
           "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  (YIN-YANG)
  (while (not (or (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if (= (car GR) 5)
             (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
             (if (and (= (car GR) 3)
                      (> (getvar "CVPORT") 1)
                      (setq P (cadr GR) PX (car P) PY (cadr P))
                      (equal (list PX PY) '(4 4) 4.48)
                      (or (< (- PX (setq X (fix PX))) 0.48)
                          (< (- (setq X (fix (1+ PX))) PX) 0.48))
                      (or (< (- PY (setq Y (fix PY))) 0.48)
                          (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                      (PUNT))
                 (progn
                   (setq Q ())
                   (CLIC)
                   (if POST
                       (if V ; (CLIC) en una casella plena.
                           (if (member P FOTO1)
                               (progn
                                 (command "ESPACIOP" "BORRA" "LT" ""
                                          "DESHACER" "M"
                                          "INSERT" "C" "0,0" "" "" "")
                                 (RETOL-2 "Anul-lant" "aquesta" "casella,"
                                          "caldrà" "recalcular" "solucions"
                                          "compatibles" "i tornar a" "esperar...")
                                 (if FOTO2
                                     (progn
                                       (prompt "\n.\n.")
                                       (SEGUIR T)
```

```
                        (command "DESHACER" "R" "DESHACER" "M"
                                 "INSERT" "C" "0,0" "" "" "")
                        (RETOL-2 "...i, si surt" "\"EL SUDOKU"
                                 "JA TÉ UNA" "SOLUCIÓ" "ÚNICA\", mai"
                                 "no sabreu" "si sobren" "caselles"
                                 "plenes.")))
                    (initget "Si No")
                    (setq CONF (getkword (strcat "\n.\n.\nConfirmeu "
                                                 "l'anul·lació "
                                                 "(S/N)? <N>: "))
                          CONF (if CONF CONF "No"))
                    (if (= CONF "Si")
                        (setq POST ())
                        (progn
                          (command "DESHACER" "R" "UY" "ESPACIOM")
                          (CLIC))))
                  (progn
                    (setq I (assoc P S-V*V)
                          J (reverse(cdr (member I (reverse S-V*V))))
                          S-V*V (append J (cdr (member I S-V*V)))
                          I (caar S-V*V)
                          J (nth (car I) (nth (cadr I) **9*9**)))
                    (NUMCOMPAT I J ())))
              (NUMCOMPAT P N T)) ; (CLIC) en una casella buida.
(if (not POST)
    (progn
      (if (and V (member P FOTO1))
          (setq POST T KKK ())
          (if (not V) (SUD-MIN **V*V** ())))
      (if POST
          (progn
            (if V
                (command "DESHACER" "R")
                (command "ESPACIOP" "BORRA" "LT" ""))
            (command "DESHACER" "M"
                     "INSERT" "C" "0,0" "" "" "")
            (RETOL-2 "Determinar" "quantes" "solucions"
                     "normalitza-" "des són" "compatibles"
                     "pot trigar" "hores." "Espereu...")
            (PERFIL-V*V)
            (COMPAT-PERFILS)
            (command "DESHACER" "R" "UY" "ESPACIOM")))))
(if POST
    (if (> KK 1)
        (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                        "normalitzades heu d'omplir més "
                        "caselles:\n"))
        (progn
          (if (not KKK) (setq FOTO1 (FOTO-V*V) FOTO2 T KKK T))
          (prompt (strcat "\n.\nEL SUDOKU JA TÉ UNA SOLUCIÓ "
                          "NORMALITZADA ÚNICA.\nOmplint més "
                          "caselles el fareu més fàcil ("
                          "<Intro> o < > per acabar)")))))))))
```

Condicions mínimes

Si ara considerem totes les situacions des de les quals tard o d'hora hem d'anar a raure als processos qualificats amb el malnom de "maquinària pesant" (**RE-MEMBER** en la fase prèvia **OMPLE-SUDOKU** de l'opció A i **COMPAT-PERFILS** en la final **FES-PUBLIC**), pels llargs temps d'espera a què poden donar lloc, la màxima prioritat serà evitar que es posin en marxa abans de ser imprescindibles, perquè fer-ho prematurament (és a dir, abans que es donin les condicions d'existència d'una solució única) no només significarà trobar-se amb pauses inesperades quan encara no tocava sinó que aquestes pauses arribin a ser llarguíssimes: de fet, tant més llargues quantes més caselles buides hi hagi en el moment de posar-les en marxa. Per això (al marge de dispositius addicionals que en l'opció A ens haurien de posar sobre avís que ja no n'hi ha prou a utilitzar les regles de joc primàries a l'hora d'omplir caselles), té tanta importancia poder enunciar unes condicions que, sense entorpir massa els processos en què les haguem d'aplicar, s'ajustin amb eficàcia a un perfil mínim de SUDOKU-PROBLEMA que l'autor no ha trobat enlloc, sense galga o amb una galga fina. Cal que l'usuari de l'opció A estigui assabentat de la conveniència d'anar omplint de forma dispersa, si més no en començar, i no concentrar-se d'entrada en un únic requadre 9×3 o 3×9, per no activar **POST**, mitjançant **T+D<12**, amb les primeres sis caselles emplenades o unes poques més (en l'últim capítol, *Consells pràctics per evitar una executio precox*, intentarem de donar-li unes pautes i, si s'escau, ja recordarem la recomanació). Però, si aconsegueix evitar de caure en aquest parany, ¿en quin moment s'haurà de recórrer a la maquinària pesant? Si el desencadenant és un criteri d'ocupació mínima, per sota de la qual no hi pot haver SUDOKU-PROBLEMA, tindrà sentit aplicar-lo a **COMPAT-PERFILS**, però fer-ho a **RE-MEMBER** sembla gratuït. Doncs bé: sí que ho és, d'arbitrari, però mentre en el camí no aparegui cap raó en contra, també l'usarem provisionalment com a llindar d'aplicació d'aquesta funció.

Com que a hores d'ara el lector ja haurà copsat molts dels tics (fílies i fòbies) que conformen el *modus operandi* de l'autor, no l'estranyarà que aquest confessi que no se li ha ocorregut cap altra cosa que examinar la pila de fulls de diari amb sudokus per resoldre (i que, un cop resolts, llençava) que guardava des de feia mesos, per veure quin tenia menys caselles plenes. Ni li va passar pel cap cercar a Google respostes que algú amb més solvència en la matèria hagués pogut donar a les dues preguntes que es feia, que eren:

- Quin es el mínim nombre de caselles que cal emplenar en un sudoku perquè pugui tenir solúció única?
- Quin és el mínim nombre de caselles ocupades d'un requadre 9×3 o 3×9 perquè un sudoku pugui tenir solució única?

El sudoku més escanyolit que va trobar a la pila esmentada va ser l'autoqualificat "difícil" del diari LA VANGUARDIA del 25-02-08, amb només 23 caselles plenes, i un requadre 9×3 i un altre 3×9 amb 6. La segona característica l'havia observada a d'altres, però de 23 caselles només n'havia trobat aquell, i així va passar més de mig any en el quasi convenciment que aquesta xifra en representava el llindar. L'altre criteri (que els requadres 9×3 o 3×9 havien de tenir 6 caselles plenes, si més no) es va esmicolar abans. Encara en data 2-07-08, l'autor va tenir un ensurt que de seguida es va transformar en triomf i confirmació d'allò que sostenia: en el diari de distribució gratuïta METRO, no a la pàgina habitual on surten sudokus més aviat senzills (però de resolució còmoda per l'amplitud de les caselles) sinó a l'anunci de l'estrena de la pel·lícula "Las crónicas de Narnia", venien dos sudokus promocionals i un d'ells presentava el requadre 9×3 superior amb només 4 caselles plenes; es va posar a fer-lo i va descobrir que no tenia una solució única sinó cinc (val a dir que les 4 caselles grogues no resultaven afectades per la indeterminació, i que amb els seus valors constants es formava la data "1941", una de les dues endevinalles en què es basava el joc). La presumpció en va sortir reforçada... fins que, en el diari EL PERIÓDICO del 16-09-08, el sudoku "difícil" mostrava el requadre 9×3 central amb només 5 caselles plenes, i sols dues setmanes més tard, el 2-10-08, el diari QUÉ! (de distribució gratuita, com METRO) en duia un amb el requadre 3×9 central de també 5 caselles. Això ja era massa: va decidir agafar el toro per les banyes i esbrinar fins on podia arribar aquest degoteig. Va pensar a utilitzar la SOLUCIÓ CANÒNICA que hem presentat unes pàgines enrera (esquerra), i no li va costar massa descobrir que el requadre 3×9 central podia deixar-lo amb només 2 caselles plenes (centre) i que, posats a depurar aquest sudoku fins a eliminar la informació redundant, el total de caselles plenes podia reduir-se a menys de la meitat, passant de 56 a 24 (dreta):

```
9 1 2   3 4 5   6 7 8      9 1 2   0 0 5   6 7 8      0 1 0   0 0 5   0 7 8
6 7 8   9 1 2   3 4 5      6 7 8   0 0 0   3 4 5      6 7 0   0 0 0   0 0 5
3 4 5   6 7 8   9 1 2      3 4 5   0 0 0   9 1 2      0 0 5   0 0 0   9 0 0

8 9 1   2 3 4   5 6 7      8 9 1   0 0 0   5 6 7      8 0 0   0 0 0   0 6 0
5 6 7   8 9 1   2 3 4      5 6 7   0 0 0   2 3 4      0 0 7   0 0 0   2 0 0
2 3 4   5 6 7   8 9 1      2 3 4   0 0 0   8 9 1      0 3 0   0 0 0   0 0 1

7 8 9   1 2 3   4 5 6      7 8 9   0 0 0   4 5 6      0 0 9   0 0 0   4 0 0
4 5 6   7 8 9   1 2 3      4 5 6   0 0 0   1 2 3      4 0 0   0 0 0   0 2 3
1 2 3   4 5 6   7 8 9      1 2 3   4 0 0   7 8 9      1 2 0   4 0 0   0 8 0
```

Aquest sudoku li va inspirar un enunciat de validesa universal: EN UN SUDOKU AMB SOLUCIÓ ÚNICA, ELS REQUADRES 9×3 i 3×9 HAN DE TENIR ALMENYS DUES CASELLES PLENES I SITUADES EN DIFERENTS FILES (SI EL REQUADRE ÉS DE 9×3) O COLUMNES (SI ÉS DE 3×9). La demostració era gairebé tautològica:

- UN SUDOKU AMB UN REQUADRE 9×3 [3×9] QUE NOMÉS TINGUI DUES CASELLES PLENES, LES DUES SITUADES A LA MATEIXA FILA [COLUMNA], O NO TÉ CAP SOLUCIÓ O EN TÉ ALMENYS DUES, perquè si n'hi ha una també serà solució la variant obtinguda permutant les files [columnes] que resten buides en el sudoku.
- UN SUDOKU AMB UN REQUADRE 9×3 [3×9] QUE NOMÉS TINGUI UNA CASELLA PLENA O NO TÉ CAP SOLUCIÓ O EN TÉ ALMENYS DUES, perquè si n'hi ha una també serà solució la variant obtinguda permutant les files [columnes] que resten buides en el sudoku.
- UN SUDOKU AMB UN REQUADRE 9×3 [3×9] SENSE CASELLES PLENES O NO TÉ CAP SOLUCIÓ O EN TÉ ALMENYS SIS, perquè si n'hi ha una també seran solucions les cinc variants obtingudes permutant les files [columnes].

Un dels dos llistons havia caigut i, si més no, savíem que ja no podia caure més.

I, de propina, un altre enunciat de validesa universal: EN UN SUDOKU AMB SOLUCIÓ ÚNICA, ELS REQUADRES 9×3 i 3×9 HAN DE TENIR ALMENYS DUES FILES (SI EL REQUADRE ÉS DE 9×3) O COLUMNES (SI ÉS DE 3×9) OCUPADES, ÉS A DIR, TOTALMENT O PARCIAL PLENES. De fet, els dos primers dels tres punts precedents no eren sinó casos particulars del principi següent:

- UN SUDOKU AMB UN REQUADRE 9×3 [3×9] QUE NOMÉS TINGUI UNA FILA [COLUMNA] OCUPADA, O NO TÉ CAP SOLUCIÓ O EN TÉ ALMENYS DUES, perquè si n'hi ha una també ho haurà de ser la variant obtinguda permutant les files [columnes] que resten buides en el sudoku.

L'altra acotació, però, semblava confirmar-se en no haber trobat cap sudoku en què el total de caselles plenes baixés de 23... fins que un dia, quan *grosso modo* el programa semblava funcionar i ja només quedava polir qüestions menors, entre elles aquesta del perfil del sudoku menys poblat, va decidir provar fortuna a Google. Tot i que sigui pura anècdota, és curiós de veure com no va saber trobar-ho a la primera (cercant directament per "sudoku-minimo", per exemple) sinó d'esquitllada. Amb la vaga idea de pescar alguna entrada del tipus "sudoku más corto" o "sudoku más vacío", si cercava per "sudoku-mas", i de passar-se després a "sudoku-menos", es va trobar un suggerent "AI Escargot, el Sudoku más difícil del mundo...", en què se l'informava que un matemàtic finlandès anomenat Arto Inkala havia proposat el sudoku teòricament més difícil de resoldre... manualment, és clar (seguint el mètode habitual de rastreig de requadres 9×3 i 3×9 per detectar les repeticions simples d'un valor i acotar-ne la tercera aparició). No va resistir la temptació de posar a prova l'eficàcia de l'opció A en el seu vessant espuri de verificadora de sudokus presumptament ben plantejats, i va comprovar que no n'hi havia per a tant: després d'emplenar les 23 caselles amb els valors proposats va passar a les buides, entrant els valors únics permesos per **MASCARA**: el de la casella 1,1 (8) va trigar força a aparèixer; el de la 2,1 (9), menys de la meitat; després de saltar-se les caselles 4,1, 5,1 i 6,1, perquè en pertànyer a un requadre 3×3 buit podien comportar llargs temps d'espera, el de la 8,1 (5) no va arribar a la dècima part; el de la 9,1 (4), pocs segons; el de la 2,1 (2) va ser pràcticament instantani i, després d'introduir-lo, la resta es va omplir automàticament per obra i gràcia d'**ACTUALITZA-PLUS**. L'autor era conscient que l'algorisme autorecursiu de **RE-MEMBER** actuava cegament i que els temps no donaven la mesura del grau de dificultat d'una resolució manual, però que aquesta funció només hagués hagut d'intervenir per emplenar 5 caselles i la resta fos trivial (en posar-s'hi ell, la va resoldre en 18 minuts, tot i ser dolentot) autoritzava a opinar que potser "Escargot" tampoc no era res de l'altre món.

Però seguim. Enmig de totes aquestes cabòries, l'autor se n'havia anat del lloc http://www.noticiasdot.com/publicaciones/2006/1106/0811/noticias081106/noticias081106-136.htm on l'havia dut l'entrada escollida i, tractant de recuperar-lo, no es va adonar que en prenia una altra de molt semblant que no el va dur al mateix lloc sinó a http://www.sofoca.cl/pebre/2007/11/10/el-sudoku-mas-dificil-del-mundo/. També s'hi oferia el sudoku de Inkala, però abans, com de passada, hi havia el comentari següent." Minimum Sudoku hizo una recopilación de 47.442 sudokus que sólo te dan diecisiete números: es **la cantidad mínima** necesaria para poder completar el juego". Un cop refet del calfred que li va sacsejar tot el cos, va tornar a llegir la frase, es va armar de valor, va picar l'hipertext i va aterrar a http://people.csse.uwa.edu.au/gordon/sudokumin.php, un document que, com aquest, està sotmès a les condicions de Creative Commons, porta per títol el mateix text subratllat i té per autor Gordon Royle, de The University of Western Australia: hi presenta 47.793 sudokus bàsics (de cada un dels quals se'n pot obtenir 940.584.959 més, per aplicació de les transformacions que aquí hem anomenat NV, TR, F1, C1, F3 i C3) que tenen en comú tenir 17 caselles plenes. Doncs bé: sembla que ara per ara aquest és el límit obtingut per via empírica i acceptat provisionalment... mentre ningú no en trobi algun de només 16 caselles o tanqui el debat de forma definitiva demostrant teòricament on se situa el llindar.

Així doncs, si recapitulem ens trobarem amb tres mínims: dos per a cada requadre 9×3 o 3×9 (2 caselles plenes i 2 files/columnes ocupades) i un per al conjunt del sudoku (17 caselles plenes). També recordareu que encara en consideràvem un altre, a càrrec de la funció **T+D<12**, però aquest era un filtre només aplicable en la fase prèvia **OMPLE-SUDOKU** de l'opció A (construir la solució pas a pas, operació en què calia detectar a partir de quin moment s'havia de fer prospectiva per garantir la correcció de la jugada, per després decidir-ne la part pública), complementant el filtre **SUD-MIN** que ara ens ocupa: aquest se situarà en la mateixa **OMPLE-SUDOKU**, entre **TECLA-M** (determinació de la casella que anem a emplenar) i **MASCARA** (on ja cal saber si només considerem les regles primàries de compatibilitat o haurem de recórrer a **RE-MEMBER**), mentre que **T+D<12** l'havíem situat en la funció **ACTUALITZA**. Però després d'haver obtingut la SUDOKU-SOLUCIÓ amb A, B o C, **SUD-MIN** tornarà a aparèixer en la composició del SUDOKU-PROBLEMA a partir de la solució, a càrrec de **FES-PUBLIC** (en els capítols precedent i següent), quan el repte no és assegurar la compatibilitat dels valors de les caselles (perquè l'únic valor possible ja n'és de compatible, per definició) sinó saber quan només queda una solució compatible.

Abans de centrar-nos en el filtre **SUD-MIN**, però, i ja que parlàvem del seu context en el programa, convé parlar de tres decisions que calia prendre per fer rutllar el dispositiu. La primera és una peculiaritat dels dos accessos, que de moment ens limitarem a enunciar però que no trigarem a justificar: des d'**OMPLE-SUDOKU** l'accés adoptarà la forma **(SUD-MIN **9*9** T)** i, des de **FES-PUBLIC**, **(SUD-MIN **V*V** ())**; si no cal aclarir què representa el primer argument, del segon direm que correspon a la variable binària **PREVI**, la missió de la qual té a veure amb una qüestió que tractarem tot seguit, però que no queda del tot resolta amb el tracte que hi rep. Aquesta segona qüestió és la localització del primer accés (fa poc hem dit que se situa entre els accessos a **TECLA-M** i a **MASCARA**), quan tot feia preveure que **SUD-MIN** s'ubicaria en **ACTUALITZA**, després de **T+D<12**, com mostra la línia subratllada:

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L LL NOU PLUS E F)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E)
                 LL (cons L LL)))
         (setq A-9*9 (reverse LL) A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST) (progn (T+D<12) (setq NOU POST)))
         (if (not POST) (SUD-MIN A-9*9))
         (if REAL
             (progn
               (ACTUALITZA-PLUS T)
               (if (not PLUS)
                   (progn
                     (setq A-9*9 (TRANSPOSAR A-9*9))
                     (ACTUALITZA-PLUS ())
                     (setq A-9*9 (TRANSPOSAR A-9*9))))
               (if (and PLUS NOU) (setq POST () NOU ())))))
         (setq PLUS (not PLUS)))
  (list A-9*9 A-LLL))
```

Doncs, no. Com veurem, resulta més senzill d'aplicar a **SUD-MIN** les condicions de no activació de **POST** (condicions que es refereixen exclusivament a l'ocupació, a diferència de les de **T+D<12**, que no són alienes als valors), seguint un esquema **(progn (setq POST T) (if (or (...) (...) ... (...)) (setq POST ())))**, i **MASCARA** haurà de recórrer a **RE-MEMBER** en el mateix moment d'activació de la casella que provoca l'incompliment d'alguna d'aquestes condicions **(not POST)**, a diferència de de **T+D<12**, en què l'activació de **POST** indica que **RE-MEMBER** haurà d'intervenir en la casella següent i successives. Potser algun lector meticulós estarà pensant a avançar-se a l'autor per advertir que l'expressió que cal traslladar d'**ACTUALITZA** a **FES-SUDOKU** i situar entre **(TECLA-M)** i **(MASCARA (ELEMENT LLL))** no serà ben bé **(if (not POST) (SUD-MIN A-9*9 T))** sinó **(if (not POST) (SUD-MIN **9*9** T))**. Però tampoc: anant a la tercera qüestió, ha de ser **(if (not POST) (SUD-MIN NOPLUS T))**. Perquè no podem passar per alt que, amb sols quatre caselles ocupades (si ho estan amb el mateix valor, i dues pertanyen a un mateix requadre 9×3 i les altres dues a un mateix requadre 3×9), **ACTUALITZA-PLUS** ja pot actuar emplenant forats. Ni que, en la mesura que els anomenats forats tapats són caselles de valor predeterminat (després d'ocupada la casella desencadenant), haurem de desestimar-los a l'hora de fer el recompte de caselles plenes a **SUD-MIN**: amb independència que apareguin en pantalla mentre dura el procés d'emplenament, són informació redundant que no ha de formar part del SUDOKU-PROBLEMA; molt probablement, algun dels sudokus mínims catalogats por Gordon Royle predetermina el valor d'algun altra casella, pero si la incloguéssim ja no seria mínim. Per això haurem de mantenir una pseudomatriu 9×9 **NOPLUS** paral·lela a ****9*9**** (seran iguals, tret dels emplenaments automàtics a càrrec d'**ACTUALITZA-PLUS**) i incorporar en **ACTUALITZA** una variable binària **PPLUS**:

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L LL NOU PLUS PPLUS E F)
  (while (not PLUS)
         (setq LL () X (car A-P) Y (cadr A-P) J -1)
         (foreach E A-9*9
           (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E)
                          LL (cons L LL)))
         (setq A-9*9 (reverse LL)
               A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST) (progn (T+D<12) (setq NOU POST)))
         (if (not (or POST PPLUS))
             (progn
               (setq LL () J -1)
               (foreach E NOPLUS
                 (setq J (1+ J) L (if (= J Y) (subst A-N A-P E) E)
                                LL (cons L LL)))
               (setq NOPLUS (reverse LL))))
         (if REAL
             (progn
               (ACTUALITZA-PLUS T)
               (if (not PLUS)
                   (progn
                     (setq A-9*9 (TRANSPOSAR A-9*9))
                     (ACTUALITZA-PLUS ())
                     (setq A-9*9 (TRANSPOSAR A-9*9))))
               (if (and PLUS NOU) (setq POST () NOU ()))))
         (setq PLUS (not PLUS) PPLUS T))
  (list A-9*9 A-LLL))
```

És clar que, repetint-se ara el mateix procediment d'actualització, en totes les iteracions **(while ...)** amb **A-9*9** i només en la primera (**PPLUS** = **nil**) amb **NOPLUS**, convindrà constituir-lo en una funció independent que anomenarem **ACT-9*9**:

```
(defun ACT-9*9 (N P L1 / L2)
  (reverse (foreach F L1 (setq L2 (cons (if (member P F) (subst N P F) F) L2)))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L LL NOU PLUS PPLUS E F)
  (while (not PLUS)
         (setq X (car A-P) Y (cadr A-P)
               A-9*9 (ACT-9*9 A-N A-P A-9*9)
               A-LLL (ACT-LLL X Y A-LLL))
         (if (not POST) (progn (T+D<12) (setq NOU POST)))
         (if (not (or POST PPLUS))
             (setq NOPLUS (ACT-9*9 A-N A-P NOPLUS)))
```

```
        (if REAL
            (progn
              (ACTUALITZA-PLUS T)
              (if (not PLUS)
                  (progn
                    (setq A-9*9 (TRANSPOSAR A-9*9))
                    (ACTUALITZA-PLUS ())
                    (setq A-9*9 (TRANSPOSAR A-9*9))))
              (if (and PLUS NOU) (setq POST () NOU ()))))
        (setq PLUS (not PLUS) PPLUS T))
  (list A-9*9 A-LLL))
```

Per deixar ja aquesta funció (de moment), direm que inicialment s'havia considerat la possibilitat de guardar en la llista **XYPLUS** les coordenades dels forats tapats, com a alternativa a la pseudomatriu **NOPLUS** paral·lela a ****9*9****, però l'estalvi de memòria dedicada a variables no compensava la complicació addicional de **SUD-MIN**, obligada al recompte X/Y per excloure les coincidències del recompte d'ocupació.

De tota manera, no únicament convé filar prim per no posar en marxa la maquinària pesant abans d'hora, sinó que l'usuari tingui un cert marge de maniobra per poder retardar aquest moment tot el que sigui possible, a base d'emplenar les caselles previstes però podent-ho fer en un ordre o en un altre per tal que les condicions per a la intervenció de **RE-MEMBER** o **COMPAT-PERFILS** triguin a produir-se i/o que, quan això s'esdevingui, el processat duri menys. En l'últim capítol donarem unes recomanacions pràctiques orientades cap aquest fi, però ara caldria afinar un pèl més, no tant per reduir la galga (que també) sinó per augmentar l'esmentat marge de maniobra: l'exigència que tot requadre 9×3 [3×9] tingui ocupades dues files [columnes] ens brinda un arma poderosa (deixar per a gairebé el final un requadre amb una sola alineació) però no massa flexible, i convindria complementar-la amb algun altre recurs.

Una primera acotació podria consistir a no lligar-se a dues condicions extremes que mai no es presentaran plegades. Si accediu a l'article citat de Gordon Royle i piqueu sobre l'hipertext *Download the file containing all of them!* per copiar-vos l'arxiu SUDOKU17.txt, us trobareu que, dels 47.793 sudokus-tipus no n'hi ha cap amb requadres 9×3 o 3×9 de només 2 caselles plenes, però n'hi ha 89 amb requadres de sols 3 (com els dos primers de la llista) i també n'hi ha de 4, 5, 6, 7, 8 i 9. Com podeu imaginar, l'autor no els ha comptat manualment sinó que ha improvisat la funció *ad hoc* **C:GORDON1**, que podeu veure al final del capítol, per tal de fer-ho. O sigui que en el cas de sudokus de 17 caselles podem assegurar que els requadres 9×3 i 3×9 han de tenir almenys 3 caselles plenes, però sabent que altres sudokus poden tenir-ne de només 2 i no havent-ne trobat cap de 18 no redundant (animem qui vulgui, a seguir l'exemple de Gordon Royle i posar-se a fer-ne un cens exhaustiu), ens curarem en salut suposant que de 18 cap amunt això ja pot passar. Tot plegat, podríem enunciar-ho així: "Perquè un sudoku disposi de solució única és condició necessària que els requadres 9×3 tinguin almenys 2 files ocupades, els requadres 3×9 tinguin almenys 2 columnes ocupades i que, una de dues: que tingui almenys 17 caselles plenes i tots els requadres 9×3 i 3×9 en tinguin almenys 3, o que tingui almenys 18 caselles plenes i tots els requadres 9×3 i 3×9 en tinguin almenys 2" (tot això, suposant que no hi hagi més sudokus de 17 valors que els 47.793 trobats per Royle, que potser n'hi ha més i algun d'aquests per descobrir té un requadre de només 2). No és gran cosa, però qui no es conforma és perquè no vol.

Una altra es podria inferir d'observacions concretes que, en rigor, no autoritzen cap generalització:

- Sobre la base de la SOLUCIÓ CANÒNICA, el sudoku mínim d'abans, amb requadre 3×9 de 2 caselles plenes, i els de 3 i 4 que presentem ara (esquerra i centre, tots dos redundants) tenen en comú la posició central dels requadres 9×3 i 3×9 menys poblats. La població del primer és 6, 7 i 8 respectivament, de forma que la suma de caselles emplenades dels requadres horitzontal i vertical més despoblats dóna 8, 10 i 12. Adoneu-vos que parlem de la suma dels dos cardinals i no del nombre de caselles plenes en el conjunt de 45 format per la unió d'un requadre 9×3 i un de 3×9: això significa que aquelles que se situïn en el requadre intersecció, de 3×3, es compten dues vegades. També podeu comprovar que, si anul·lem la casella 3,5 (7) o la 7,5 (2) de l'últim esmentat (centre), en sortiran sudokus estrictes (solució única) en què la suma de caselles públiques dels requadres centrals és 7 + 4 = 11 (no els hem representat, i del de la dreta en parlarem de seguida).

```
0 1 0   0 0 5   0 7 8      0 1 0   0 0 5   0 7 8      0 1 0   0 0 5   0 7 8
6 7 0   0 0 0   0 0 5      6 7 0   0 0 0   0 0 5      6 7 0   0 0 0   0 0 5
0 0 5   0 0 0   9 0 0      0 0 5   0 0 0   9 0 0      0 0 5   0 7 0   9 0 0

8 0 0   0 0 0   0 6 0      8 0 0   0 0 4   0 6 0      8 0 0   0 0 0   0 0 0
0 0 7   0 9 0   2 0 0      0 0 7   0 0 0   2 0 0      0 0 7   0 0 0   2 0 0
0 3 0   0 0 0   0 0 1      0 3 0   5 0 0   0 0 1      0 0 0   0 0 0   0 0 1

0 0 9   0 0 0   4 0 0      0 0 9   0 0 0   4 0 0      0 0 9   0 2 0   4 0 0
4 0 0   0 0 0   0 2 3      4 0 0   0 0 0   0 2 3      4 0 0   0 0 0   0 2 3
1 2 0   4 0 0   0 8 0      1 2 0   4 0 0   0 8 0      1 2 0   4 0 0   0 8 0
```

- Amb una altra funció *ad hoc* **C:GORDON2** (que, com **C:GORDON1**, trobareu al final del capítol) hem descobert que, dels 47.793 sudokus-tipus no n'hi ha cap en què la suma de les caselles plenes dels requadres 9×3 i 3×9 menys poblats sigui menor que 7 (de 7 n'hi ha 7, i també n'hi ha de 8, 9 i 10, que completen els 47.793).

- En els dos sudokus localitzats a la premsa amb requadres 9×3 o 3×9 de 5 caselles plenes, aquestes sumes donaven 12 i 13.

- En tots els altres sudokus, mai no hi sortien requadres 9×3 o 3×9 amb menys de 6 caselles plenes cadascun, de manera que la suma esmentada mai no baixava de 12.

A la vista d'això, doncs, ¿seria molt agosarat d'afirmar que la suma de caselles plenes d'un requadre 9×3 i d'un altre 3×9 no pot ser inferior a 7? Potser sí, però l'autor assumeix el risc d'haver d'entonar el *mea culpa* i rectificar quan calgui. El que sí que s'atreveix a assegurar és que l'adopció d'aquesta tesi permetrà en bastants casos de retardar l'activació de **COMPAT-PERFILS** i augmentar el repertori de maniobres per retardar-la encara més. Per veure com es pot treure partit de la nova variable seguirem amb la SOLUCIÓ CANÒNICA, però en comptes d'utilitzar algun dels sudokus precedents amb 24, 25 o 26 caselles plenes, experimentarem amb un de nou, estricte i amb 24 (dreta): molt semblant a l'altre amb 24 (el tercer de dues pàgines enrera), aquest té una disposició que s'aproxima més a la simetria central (en particular pel que fa als braços de la creu formada pels requadres centrals de 9×3 i 3×9) i podríem obtenir-lo buidant les caselles 2,4 (3) i 8,6 (6), i omplint les 5,3 (2) i 5,7 (7). Fora de la llista de Gordon Royle, aquests dos són els que més s'acosten al límit d'aplicació del requisit que postulem, perquè la suma de les caselles plenes dels requadres centrals dóna 8. Tot apunta a la hipòtesi que, únicament en els esmentats 7 sudokus de 17 caselles, aquesta suma dóna 7 i que, a mesura que considerem sudokus més poblats, si no puja la suma, almenys no baixarà.

Ara que sabem què cal omplir, ho farem seguint seqüències diferents i en cada una veurem fins on ens permet arribar **SUD-MIN** abans que es posi en marxa el dispositiu **PERFIL-V*V** + **COMPAT-PERFILS**, valorant si la incorporació de la condició addicional endarrereix o no aquest succés: omplint de manera rutinària, d'esquerra a dreta i de baix a dalt, de poder emplenar 18 caselles passarem a emplenar-ne 22 (esquerra, en aquesta pàgina i en la següent); procedint d'igual manera en les set primeres files però seguint amb la 9ª i la 8ª, n'emplenarem 19 sense aplicar el criteri i 23 aplicant-lo (centre, en aquesta pàgina i en la següent); procedint igualment però saltant-nos el segon requadre 9×3 i deixant-lo per al final, en el primer cas emplenarem 22 caselles (dreta, en aquesta pàgina i en la següent) i en l'altre totes les 24; procedint igualment pel que fa als requadres però atacant la fila 5ª abans que la 4ª i 6ª, en el primer cas emplenarem 23 caselles (dreta, en la pàgina següent) i en l'altre totes les 24.

```
0 0 0   0 0 0   0 0 0      0 1 0   0 0 5   0 7 8      0 1 0   0 0 5   0 7 8
6 0 0   0 0 0   0 0 0      6 0 0   0 0 0   0 0 0      6 7 0   0 0 0   0 0 5
0 0 5   0 7 0   9 0 0      0 0 0   0 0 0   0 0 0      0 0 5   0 7 0   9 0 0

8 0 0   0 0 0   0 0 0      8 0 0   0 0 0   0 0 0      0 0 0   0 0 0   0 0 0
0 0 7   0 0 0   2 0 0      0 0 7   0 0 0   2 0 0      0 0 7   0 0 0   0 0 0
0 0 0   0 0 0   0 0 1      0 0 0   0 0 0   0 0 1      0 0 0   0 0 0   0 0 1

0 0 9   0 2 0   4 0 0      0 0 9   0 2 0   4 0 0      0 0 9   0 2 0   4 0 0
4 0 0   0 0 0   0 2 3      4 0 0   0 0 0   0 2 3      4 0 0   0 0 0   0 2 3
1 2 0   4 0 0   0 8 0      1 2 0   4 0 0   0 8 0      1 2 0   4 0 0   0 8 0
```

```
0 1 0   0 0 5   0 0 0       0 1 0   0 0 5   0 7 8       0 1 0   0 0 5   0 7 8
6 7 0   0 0 0   0 0 5       6 7 0   0 0 0   0 0 5       6 7 0   0 0 0   0 0 5
0 0 5   0 7 0   9 0 0       0 0 5   0 7 0   0 0 0       0 0 5   0 7 0   9 0 0

8 0 0   0 0 0   0 0 0       8 0 0   0 0 0   0 0 0       0 0 0   0 0 0   0 0 0
0 0 7   0 0 0   2 0 0       0 0 7   0 0 0   2 0 0       0 0 7   0 0 0   2 0 0
0 0 0   0 0 0   0 0 1       0 0 0   0 0 0   0 0 1       0 0 0   0 0 0   0 0 1

0 0 9   0 2 0   4 0 0       0 0 9   0 2 0   4 0 0       0 0 9   0 2 0   4 0 0
4 0 0   0 0 0   0 2 3       4 0 0   0 0 0   0 2 3       4 0 0   0 0 0   0 2 3
1 2 0   4 0 0   0 8 0       1 2 0   4 0 0   0 8 0       1 2 0   4 0 0   0 8 0
```

Amb aquestes incorporacions l'enunciat quedaria així: "Perquè un sudoku disposi de solució única és condició necessària que els requadres 9×3 tinguin almenys 2 files ocupades, que els requadres 3×9 tinguin almenys 2 columnes ocupades, que la suma del nombre de caselles plenes de qualsevol requadre 9×3 i el de qualsevol requadre 3×9 sigui almenys 7 i que, una de dues: que el sudoku tingui almenys 17 caselles plenes i tots els requadres 9×3 i 3×9 en tinguin almenys 3, o que tingui almenys 18 caselles plenes i tots els requadres en tinguin almenys 2". I parem esment en l'últim subenunciat, on no s'hi val a tirar pel dret i pretendre simplificar dient que "perquè un sudoku disposi de solució única és condició necessària que ... i que la suma del nombre de caselles plenes del sudoku i del que tingui qualsevol requadre 9×3 o 3×9 sigui almenys 20 (17 + 3 = 18 + 2 = 20), perquè això seria tant com acceptar que un sudoku amb 19 o més caselles plenes i un requadre amb només una, un sudoku amb 20 o més caselles plenes i un requadre que no en tingui cap, o bé un sudoku amb menys de 17 caselles plenes i tots els requadres amb més de 3, poden tenir solució única. Ben diferent és recórrer al complement lògic i dir: "Per poder afirmar que un sudoku encara no té solució única, és condició suficient que s'esdevingui algun dels successos següents: que algun requadre 9×3 tingui 2 files buides; que algun requadre 3×9 tingui 2 columnes buides; que la suma del nombre de caselles plenes d'algun requadre 9×3 i d'algun de 3×9 doni menys de 7; que tingui menys de 17 caselles plenes; que algun requadre 9×3 o 3×9 en tingui menys de 2; que tingui menys de 18 caselles plenes i que algun requadre 9×3 o 3×9 en tingui menys de 3". Si les condicions s'avaluen en aquest mateix ordre, llavors l'última sí que equival a enunciar "que la suma del nombre de caselles plenes del sudoku i el d'algun requadre 9×3 o 3×9 sigui menor que 20", perquè havent deixat a fora prèviament els sudokus amb menys de 17 ocupacions y els que tenien requadres amb menys de 2, això només pot referir-se a sudokus de 17 amb algun requadre de 2.

La funció **SUD-MIN** es basarà en aquest enunciat complementari, que resulta còmode perquè jugant amb la variable binària **POST**, activada a l'inici, no costarà gens desactivar-la quan es produeixi algun d'aquests successos, per interrompre'n la verificació, i només quan a la sortida encara sigui **POST** es posarà en marxa algun dels processos que hem qualificat de "maquinària pesant". Pel que fa a l'últim succés (última línia de codi), adoptarem la formulació més simple amb què acabàvem el paràgraf precedent (per bé que a continuació oferim tres fórmules equivalents, caracteritzades com a comentaris):

```
(defun SUD-MIN (9L PREVI / J0 J1 J2 J3)
  (setq I 0 J 0 K 0 POST T)
  (foreach LL (list 9L (TRANSPOSAR 9L))
    (if (and J1 POST (< I 17)) (setq POST ()))
    (setq J3 1)
    (if POST
        (foreach L LL
          (if POST
              (progn
                (foreach E L
                  (if (or (atom E) (and PREVI (equal E (list X Y))))
                      (progn
                        (if (or (and (not J1) (= J3 1))
                                (and J1 (not J2) (> J3 1)))
                            (setq I (1+ I)))
                        (setq J (1+ J)))))
                (setq J0 (cons J J0))
                (if (< K 2)
                    (setq K (1+ K))
```

```
                    (if (or (< J 2)
                            (equal J0 (list J 0 0))  ; 1ª i 2ª alineació buides.
                            (equal J0 (list J J 0))  ; 1ª i 3ª alineació buides.
                            (equal J0 (list J J J))) ; 2ª i 3ª alineació buides.
                        (setq POST ())
                        (progn
                          (if J1
                              (if J2
                                  (setq J2 (if (< J J2) J J2)
                                        POST (or (< J3 3) (>= (+ J1 J2) 7)))
                                  (if (= J3 1)
                                      (setq J2 J)
                                      (setq J1 (if (< J J1) J J1))))
                              (setq J1 J))
                          (setq J0 () J 0 K 0 J3 (1+ J3))))))))))
  (if (and POST (= I 17) (= (min J1 J2) 2)) (setq POST ())) ; Aquesta línia podia
; (if (and POST (< I 18) (< (min J1 J2) 3)) (setq POST ())) ; haver-se substituït
; (if (and POST (= (+ I (min J1 J2)) 19)) (setq POST ()))   ; per qualsevol de
; (if (and POST (< (+ I (min J1 J2)) 20)) (setq POST ()))   ; les 3 alternatives.
```

No podem deixar-ho aquí, sense aclarir abans el paper que hi juga **PREVI** (argument binari del qual havíem anunciat que adoptava els valors **T** i **nil** en els accessos des d'**OMPLE-SUDOKU** i **FES-PUBLIC**, respectivament) i la diferència entre la condició simple **(atom E)** la múltiple **(or (atom E) (and PREVI (equal E (list X Y)))**. Des de **FES-PUBLIC** l'avaluació de **SUD-MIN** es farà amb la casella emplenada (tot i ser amb un valor preestablert) perquè, quan el resultat sigui **POST** = **T**, la primera acció a emprendre serà la determinació del nombre de solucions normalitzades compatibles amb els emplenaments realitzats, procés disortadament llarg del qual ens advertirà un rètol que substitueix el *taijitu* (símbol que donava llum verda a l'activació de noves caselles), en la part central de la pantalla gràfica d'AutoCAD, i també, de tard en tard, informació sobre la marxa del procés en la finestra de text (visible en l'àrea d'ordres i, quan aquest ha acabat, ampliant el text a tota la pantalla), perquè l'usuari rebi senyals d'activitat i no tingui la sensació que l'ordinador s'ha penjat. Per contra, des d'**OMPLE-SUDOKU** n'hi ha d'haver prou a seleccionar la casella que volem emplenar, abans de saber quins valors són assignables i triar-ne un, perquè precisament l'activació de **POST** ha de marcar el pas del règim ràpid al règim lent d'aparició dels candidats admesos en el caseller central 9×9 (l'últim, acompanyat d'un advertiment que emmarca el caseller, per si no fos prou evident el canvi de ritme). És per això que la casella **E** seleccionada però encara no activada (**(equal E (list X Y))**) compti tant com les ocupades (**(atom E)**) des de **PREVI** = **T**.

Encara rai que els descobriments i les decisions descrits en les últimes 7 pàgines es van succeir ràpidament, malgrat haver començat les classes i haver-se produït la conseqüent afluixada en la dedicació al tema! Si no, en trobar-se l'autor en EL PAÍS del dia 24-10-08 un sudoku "difícil" de 23 caselles plenes, amb el requadre 3×9 central de només 5 (conjunció de circumstàmcies que tres setmanes abans encara estava convençut que constituïen el llindar absolut), potser hauria comés l'error de veure-hi una confirmació a les seves intuicions i ja no hauria anat més enllà.

Com havíem dit, aquí teniu les funcions **C:GORDON1** i **C:GORDON2**:

```
; Les ordres GORDON1 i GORDON2 serveixen per fer recomptes en l'arxiu SUDOKU17.txt
; que, obtingut des del lloc "http://people.csse.uwa.edu.au/gordon/sudokumin.php",
; cal haver copiat en algun directori de la ruta de cerca establerta. Aquest arxiu
; consta de 47.793 línies de 81 caràcters numèrics, que representen els sudokus-
; tipus amb 17 caselles inventariats pel professor Gordon Royle de The University
; of Western Australia.
; En concret, GORDON1 compta quants sudokus de la col·lecció tenen requadres 9x3 o
; 3x9 amb N caselles plenes (N entre 3 i 9 caselles):
;     N = 3:     89        N = 4: 17.113        N = 5: 46.820        N = 6: 44.682
;     N = 7: 34.307        N = 8:  1.730        N = 9:     17
; Sumen més de 47.793, perquè cada sudoku intervé en més d'un còmput: per exemple,
; un sudoku amb requadres de 9x3 de 5, 6 i 6 caselles plenes i requadres 3x9 de 4,
; 6 i 7 caselles plenes intervé en quatre d'aquests còmputs.
(defun C:GORDON1 (/ M N ARX I J K OK 9*9 9*3)
  (initget)
  (prompt "\nAssegura't que l'arxiu SUDOKU17.txt és a la ruta de cerca d'AutoCAD")
  (prompt ".\nGORDON1 comptarà quants dels 47.793 sudokus amb 17 caselles tenen")
```

```
  (setq M (getint "\nrequadres 9x3 o 3x9 amb només N caselles plenes. Entra N: ")
        N 0 ARX (open (findfile "SUDOKU17.txt") "r"))
  (while (setq OK T I "" 9*9 (read-line ARX))
         (repeat 2
           (if (and OK (numberp I))
               (progn
                 (setq I "" J "" K "")
                 (repeat 9
                   (foreach E (list 'I 'J 'K)
                     (set E (strcat (eval E) (substr 9*9 1 3)))
                     (setq 9*9 (substr 9*9 4))))
                 (setq 9*9 (strcat I J K))))
           (setq I -26)
           (repeat 3
             (if OK
                 (progn
                   (setq I (+ I 27) 9*3 (substr 9*9 I 27) J 0 K 0)
                   (repeat 27
                     (if OK
                         (progn
                           (setq J (1+ J))
                           (if (> (substr 9*3 J 1) "0")
                               (setq K (1+ K) OK (<= K M))))))
                   (if OK
                       (if (= K M) (setq OK () N (1+ N)))
                       (setq OK T)))))))
  (close ARX)
  (prompt (strcat "\nEntre els 47.793 sudokus amb 17 caselles, n'hi ha " (itoa N)
                  "\namb requadres 9x3 o 3x9 de " (itoa M) " caselles plenes."))
  (princ))
; En concret, GORDON2 compta en quants sudokus de la col·lecció la quantitat de
; caselles plenes del requadre 9x3 menys ocupat i la del requadre 3x9 menys ocupat
; sumen N (N entre 7 i 10 caselles):
;       N =  7:      7
;       N =  8:  1.258
;       N =  9: 15.930
;       N = 10: 30.598
;       --------------
;       Sumen   47.793
(defun C:GORDON2 (/ M N N1 N2 ARX I J K OK 9*9 9*3)
  (initget)
  (prompt "\nAssegura't que l'arxiu SUDOKU17.txt és a la ruta de cerca d'AutoCAD")
  (prompt ".\nGORDON2 comptarà en quants dels 47.793 sudokus de 17 caselles,\n")
  (setq M (getint "les dels requadres 9x3 i 3x9 menys poblats sumen N. Entra N: ")
        N 0 ARX (open (findfile "sudoku17.txt") "r"))
  (while (setq N1 9 9*9 (read-line ARX))
         (repeat 2
           (if (< N1 9)
               (progn
                 (setq N2 N1 N1 9 I "" J "" K "")
                 (repeat 9
                   (foreach E (list 'I 'J 'K)
                     (set E (strcat (eval E) (substr 9*9 1 3)))
                     (setq 9*9 (substr 9*9 4))))
                 (setq 9*9 (strcat I J K))))
           (setq I -26)
           (repeat 3
             (setq I (+ I 27) 9*3 (substr 9*9 I 27) J 0 K 0)
             (repeat 27
               (setq J (1+ J))
               (if (> (substr 9*3 J 1) "0") (setq K (1+ K))))
             (if (< K N1) (setq N1 K))))
         (if (= (+ N1 N2) M) (setq N (1+ N))))
  (close ARX)
  (prompt (strcat "\nEntre els 47.793 sudokus amb 17 caselles, n'hi ha " (itoa N)
                  "\nen què el nombre de caselles plenes del requadre 9x3 i\n"
                  "del requadre 3x9 que en tenen menys sumen " (itoa M) "."))
  (princ))
```

Solucions compatibles ho són totes les que hi són...

En aquest capítol ens ocuparem del dispositiu que fa l'inventari de les solucions normalitzades compatibles amb la part pública de la V. E. NORMALITZADA. Recordareu que l'abandó de la utopia del capítol 1 no va suposar cap solució de continuïtat en realació a l'ús d'aquest tipus de solucions, sinó que vam passar directament de pretendre'n l'inventari exhaustiu a un de limitat a solucions compatibles amb les caselles ocupades quan un altre dispositiu (la funció **SUD-MIN** de l'últim capítol, activant la variable **POST**) ens alertés que anàvem a ultrapassar el llindar en què aquestes caselles podien constituir ja una SUDOKU-SOLUCIÓ, però sense perdre mai de vista els avantatges que estàvem segurs que comportava treballar amb solucions normalitzades. Malgrat això, com que el tàndem format per les funcions **PERFIL-V*V** i **COMPAT-PERFILS** (que té la difícil missió de detectar sobre la marxa les que són compatibles) ja és prou complicat, hem preferit de desglossar la presentació en 2 etapes: en la primera farem com si cerquéssim tota mena de solucions compatibles, i en la segona introduirem les seleccions pertinents per considerar exclusivament les normalitzades; fent-ho d'aquesta manera no només facil·litarem la comprensió del procediment sinó que quedaran més en evidència les simplificacions aplicables.

La cerca l'escometrem en tres fases:
- Inventariar les files primeres, segones... novenes compatibles amb les caselles activades.
- Limitant-nos a les files anteriors, inventariar les tríades de files compatibles entre si, adscrites a cadascun dels requadres 9×3.
- Limitant-nos a les triades precedents, inventariar les combinacions de 3 tríades compatibles entre si.

A mesura que en cada etapa s'obtingui informació quantitativa sobre els elements compatibles detectats (files, requadres 9×3 i solucions), la finestra de text que havíem dimensionat prèviament a 3 línies ens mostrarà una informació que no sempre serà possible seguir en temps real perquè circularà massa ràpidament. Doncs, ¿per què ens hi molestem?: ras i curt, per donar senyals de vida; l'espera pot ser tan llarga que qualsevol manifestació d'activitat computacional serà agraïda per un usuari que, tot i el desmoralitzador advertiment que es mantindrà tota l'estona a la zona central de l'àrea gràfica (*Determinar quàntes solucions compatibles hi ha pot trigar hores. Espereu...*), sentirà cert dessassosec en veure passar el temps. Finalment, i com a culminació del procés, la pantalla gràfica commutarà a pantalla de text, presentant tota la informació fins aleshores sols vista fragmentàriament.

La primera cosa que necessitem és disposar de la llista **L1** dels 362.880 diferents textos possibles compostos per 9 caràcters numèrics no repetits, així que haurem d'advertir l'usuari que cal tenir a mà (en la ruta de cerca d'AutoCAD) el fitxer 123456789.txt: si no el té enlloc, haurà de sortir del programa i crear-lo amb la funció **C:FES-123456789** vista en el capítol *Inventariar les solucions: un cul-de-sac*; si el té però en un directori alié a la ruta esmentada, podrà traslladar-lo o copiar-lo en un lloc més adient, i seguir amb l'execució. La funció **PERFIL-V*V**, que té per objecte obtenir la llista **LINIES-V*V** constituïda per 9 subllistes, cada una de les quals emmagatzema els textos que són compatibles amb la part pública de cada fila de ****V*V****, començarà precisament amb una crida a **C:CARREGA-123456789** (una altra funció del capítol esmentat), que transferirà els 362.880 textos a la llista **L1**. Tot i la major lentitud de treballar amb textos (cadenes de caràcters), **PERFIL-V*V** usarà aquest format perquè la funció **wcmatch** ens permetrà despatxar en una única operació la verificació de compatibilitat que, amb nombres enters, seria força més complicada. El procediment consisteix a construir el perfil **L** de cada fila de ****V*V**** i, considerant tots els elements **E** de **L1**, incorporar a la llista **LINIA-V*V** representativa d'aquesta fila aquells que encaixin en el perfil **L**, en el sentit de complir-se **(wcmatch E L)**. Prèviament a **L**, haurem hagut de formar dues llistes que reflecteixen les ocupacions de requadres 3×3 i columnes, i representen els valors exclosos per a les altres caselles d'aquests requadres 3×3 i columnes:
- **L1C** és una llista de 9 subllistes, cada una (representativa d'una fila) formada al seu torn per 9 llistes representatives de cada casella, amb valors a excloure per ser públics en el requadre 3×3 a què pertany la casella.
- **L2C** és una llista de 9 subllistes, cada una de les quals és representativa d'una columna i conté valors a excloure per ser públics en la columna en qüestió.

Aquesta funció, que correspon a la primera de les etapes enumerades, quedarà així:

```
(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V)
  (if (not L1) (C:CARREGA-123456789))
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
  (repeat 3
    (setq J (- J 9) K (+ K 3))
    (repeat 3
      (setq J (+ J 3) K (- K 3) W ())
      (repeat 3
        (setq J (- J 3) K (1+ K))
        (repeat 3
          (setq J (1+ J) V (nth J (nth K **V*V**)))
          (if (atom V) (setq W (cons V W)))))
      (cond ((= J 2) (setq A W)) ((= J 5) (setq B W)) (T (setq C W))))
    (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
  (setq L1C (reverse L1C) J -1)
  (repeat 9
    (setq J (1+ J) K -1 L ())
    (repeat 9 (setq K (1+ K) C (nth J (nth K **V*V**))
                    L (if (atom C) (cons C L) L)))
    (setq L2C (cons L L2C)))
  (setq L2C (reverse L2C) LINIES-V*V () K -1)
  (terpri)
  (repeat 9
    (setq LINIA-V*V () L "" K (1+ K))
    (mapcar '(lambda (E F G)
               (setq L (strcat L (if (or F G)
                                     (if (atom E)
                                         (itoa E)
                                         (progn
                                           (setq C "]")
                                           (foreach H (append F G)
                                             (setq C (strcat (itoa H) C)))
                                           (strcat "[~" C)))
                                     "#"))))
            (nth K **V*V**) (nth K L1C) L2C)
    (foreach E L1 (if (wcmatch E L) (setq LINIA-V*V (cons E LINIA-V*V))))
    (setq LINIES-V*V (cons (reverse LINIA-V*V) LINIES-V*V)
          *K* (itoa (length LINIA-V*V)) *KK* (cons *K* *KK*))
    (terpri)
    (prompt (strcat "\nEntre 362880 línies, de compatibles amb la " (itoa (1+ K))
                    "ª de VARIANT ESCOLLIDA n'hi ha " *K* ".")))
  (setq LINIES-V*V (reverse LINIES-V*V) L1 () *KK* (reverse *KK*))
  (terpri))
```

Algú podria pensar que encara hauríem pogut estrènyer més la selecció, introduint en totes les caselles de cada perfil **L** de fila l'exclusió dels valors públics en la fila (anàlogament a com ho fèiem per a les columnes) i suprimint de la llista **L1** els elements que ja s'haguessin incorporat a **LINIES-V*V** per ser compatibles amb alguna de les files precedents (per tenir cada cop un menor nombre de textos amb què comparar els perfils de les files), però s'equivocaria:

- Introduir en el perfil de las caselles lliures l'exclusió dels valors assignats a les ocupades no serveix de res: els elements de **L1** estan formats per caràcters no repetits i, si en el perfil de la fila hi figura una casella amb un valor, de segur que aquest valor no pot reaparèixer en una altra casella, de manera que la prevenció seria supèrflua.
- Un mateix element de **L1** pot ser compatible amb diverses files: penseu, si no, en dues files que pertanyin al mateix requadre 9×3, buides o amb totes les caselles ocupades alineades verticalment, que tindran perfils idèntics.

La segona i tercera etapa aniran a càrrec de **COMPAT-PERFILS**. Amb cada requadre 9×3 la funció començarà comparant totes les primeres files de **(car LINIES-V*V)** amb les segones de **(cadr LINIES-V*V)** i establint-ne la compatibilitat mitjançant la funció **COMPAT-2L** i, si s'escau, seguirà comparant aquestes dues amb les terceres files de **(caddr LINIES-V*V)** i establint-ne la compatibilitat mitjançant **COMPAT-3L**. D'aquest procés en resultaran llistes de primers, segons i tercers requadres 9×3 amb les 3 files compatibles entre si (i, per tant, amb la solució escollida) que s'aplegaran en la superllista **SUDOKUS-V*V** (variable que, en la tercera etapa serà reutilitzada per emmagatzemar-hi totes les solucions obtingudes, compatibles amb la escollida.

Però abans d'anar a aquesta última etapa convé aclarir el funcionament de **COMPAT-2L** i **COMPAT-3L**. **COMPAT-2L** indica si dues files són compatibles, comprovant que a cadascuna de les 9 posicions no hi figuri el mateix valor. Contra allò que pugui semblar al primer cop d'ull, no n'hi ha prou a comprovar-ne només 8, prescindint de l'última; si no, mireu les files-text **"123456789"** **"214365879"**, on els 4 primers parells de caràcters han estat permutats. Pel que fa a **COMPAT-3L**, ens indica si tres files compatibles en el sentit de **COMPAT-2L** també ho són en el sentit de no tenir cap valor repetit a cadascun dels requadres 3×3 (dit d'una altra manera, a tenir cada requadre 3×3 tots els valors **"1"**... **"9"**). Aquí sí que podrem limitar la cerca a només dos requadres 3×3, en la seguretat que, si compleixen aquest requisit, el tercer també el complirà: en el conjunt de les tres línies hi ha 3 valors **"1"**, 3 valors **"2"** ... i 3 valors **"9"**, i com que en el primer requadre n'hi ha un de cada i en el segon també, la resta (un de cada) haurà d'estar en el tercer requadre.

La tercera etapa té com a *input* l'esmentada superllista **SUDOKUS-V*V** formada per llistes de primers, segons i tercers requadres 9×3 amb les 3 files compatibles entre si, i com a *output* la superllista de les solucions obtingudes refonent els elements d'aquestes llistes compatibles entre si. En questa segona superllista, que com hem dit reutilitza el nom de la primera, les files ja no es representen amb textos formats per 9 caràcters numèrics sinó amb llistes formades pels valors numèrics corresponents. La conversió des d'un format a l'altre es deu a la funció **SUD-NUM**, però el protagonisme se l'enduu la funció predeterminada **wcmatch**, com en **PERFILS-V*V**, que en aquesta ocasióm verificarà la compatibilitat entre caràcters homòlegs de cada combinació binària i ternària de requadres 9×3 que hagin mostrat la seva compatibilitat interna. Com a funcions auxiliars hi intervenen **COLS-3L** i **PERF-COLS**: la primera converteix una tríada de files-text en una llista amb les 9 columnes de 3 caràcters corresponents a aquesta tríada; la segona converteix els elements d'aquestes llistes (aptes com a primer argument de **wcmatch**) en perfils o patrons de comparació (aptes com a segon argument), pel senzill procediment que consisteix a encadenar-los amb **"*["** per davant i amb **"]*"** per darrera, artifici que servirà per determinar si les columnes homòlogues de dos requadres 9×3 tenen algun caràcter en comú i, per extensió a les 9 columnes, si dos requadres (primer i segon; segon i tercer; primer i tercer) són compatibles.

Segurament la descripció del procés quedarà més clara amb un exemple. Suposem que una de les solucions compatibles és una vella coneguda: la SOLUCIÓ CANÒNICA (a la dreta).

9 1 2	3 4 5	6 7 8
6 7 8	9 1 2	3 4 5
3 4 5	6 7 8	9 1 2
8 9 1	2 3 4	5 6 7
5 6 7	8 9 1	2 3 4
2 3 4	5 6 7	8 9 1
7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	4 5 6	7 8 9

A **SUDOKUS-V*V** (1ª accepció) els tres requadres 9×3 seran:
("123456789" "456789123" "789123456") < **(car SUDOKUS-V*V)**
("234567891" "567891234" "891234567") < **(cadr SUDOKUS-V*V)**
("345678912" "678912345" "912345678") < **(last SUDOKUS-V*V)**
A partir d'aquí, la funció **COLS-3L** crearà **CC0**, **CC1** i **CC2**:
("147" "258" "369" "471" "582" "693" "714" "825" "936"),
("258" "369" "471" "582" "693" "714" "825" "936" "147") i
("369" "471" "582" "693" "714" "825" "936" "147" "258").
Per comprovar si el segon requadre 9×3 és compatible amb el primer caldrà que l'expressió següent sigui **nil**, és a dir, que tots els arguments **wcmatch** siguin **nil** (atenció: a **COMPAT-PERFILS**, el que aquí representem com arguments d'un operador lògic **or** són iteracions d'un bucle):

```
(or (wcmatch "258" "*[147]*") (wcmatch "369" "*[258]*") (wcmatch "471" "*[369]*")
    (wcmatch "582" "*[471]*") (wcmatch "693" "*[582]*") (wcmatch "714" "*[693]*")
    (wcmatch "825" "*[714]*") (wcmatch "936" "*[825]*") (wcmatch "147" "*[936]*"))
```

Com que, òbviament (que els textos de 3 caràcters en columna els tinguin permutats és una coincidència deguda a la singularitat de la solució escollida), aquests dos requadres són compatibles, seguirem comprovant si el tercer requadre 9×3 ho és amb el segon i el primer. Igual que abans, caldrà que l'expressió següent sigui **nil**, que representa una visió no només estàtica sinó redundant (la podríem reduir a una funció **or** amb 18 arguments, però fent-ho així reproduirem millor cada iteració):

```
(or (or (wcmatch "369" "*[147]*") (wcmatch "369" "*[258]*"))
    (or (wcmatch "471" "*[258]*") (wcmatch "471" "*[369]*"))
    (or (wcmatch "582" "*[369]*") (wcmatch "582" "*[471]*"))
    (or (wcmatch "693" "*[471]*") (wcmatch "693" "*[582]*"))
    (or (wcmatch "714" "*[582]*") (wcmatch "714" "*[693]*"))
    (or (wcmatch "825" "*[693]*") (wcmatch "825" "*[714]*"))
    (or (wcmatch "936" "*[714]*") (wcmatch "936" "*[825]*"))
    (or (wcmatch "147" "*[825]*") (wcmatch "147" "*[936]*"))
    (or (wcmatch "258" "*[936]*") (wcmatch "258" "*[147]*")))
```

La missió de la variable **S-V*V**, que en acabar **COMPAT-PERFILS** és una llista que té únicament una sublista amb la posició **P** de l'última casella activada (just abans d'executar-se **PERFIL-V*V** i **COMPAT-PERFILS**) seguida per tants valors binaris **T** com solucions compatibles **SUDOKUS-V*V** (2ª accepció) s'hagin localitzat, és portar un diari actualitzat d'emplenaments i desocupacions (activacions i desactivacions de caselles) efectuades a partir d'ara, indicant quines de les solucions **SUDOKUS-V*V** hauran passat a incompatibles o hauran recuperat la compatibilitat, però d'això en parlarem després de **PERFIL-V*V** i **COMPAT-PERFILS**, en passar a la funció **NUMCOMPAT.** Veiem ara la traducció a codi de **COMPAT-PERFILS** i les seves funcions subordinades:

```
(defun COMPAT-2L ()
  (setq K 0 OK T)
  (while (and OK (< K 9))
         (setq K (1+ K) OK (/= (substr T0 K 1) (substr T1 K 1)))))

(defun COMPAT-3L (/ M)
  (setq K 0 OK T)
  (while (and OK (< K 9))
         (setq K (1+ K) OK (/= (substr T1 K 1) (substr T2 K 1) (substr T0 K 1))))
  (if OK
      (progn
        (setq K -2)
        (repeat 2 (if OK (progn
                           (setq M "" K (+ 3 K))
                           (foreach L (list T0 T1 T2)
                             (setq M (strcat M (substr L K 3))))
                           (foreach N TOT
                             (if (and OK (not (wcmatch M N))) (setq OK ()))))))))
  OK)

(defun COLS-3L (I / C CC)
  (setq K 10)
  (repeat 9
    (setq K (1- K) C "")
    (foreach J I (setq C (strcat C (substr J K 1))))
    (setq CC (cons C CC))))

(defun PERF-COLS (CC) (mapcar '(lambda (C) (strcat "*[" C "]*")) CC))

(defun SUD-NUM (/ LN LLN)
  (foreach 3L (list T2 T1 T0)
    (foreach L (reverse 3L)
      (setq LN () K 10)
      (repeat 9 (setq K (1- K) LN (cons (atoi (substr L K 1)) LN)))
      (setq LLN (cons LN LLN)))))

(defun COMPAT-PERFILS (/ L1 LL0 LL1 L2 LL2 3L M N TERCETS-V*V CC0 CC1 CC2 CCC1
                         CCC2 TT0 TT1 TT2 TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  (repeat 9 (setq K (1+ K) TOT (cons (strcat "*" (itoa K) "*") TOT)))
  (repeat 3
    (setq LL0 (car LINIES-V*V) LL1 (cadr LINIES-V*V) LL2 (caddr LINIES-V*V)
          LINIES-V*V (cdddr LINIES-V*V) TERCETS-V*V ())
    (foreach T0 LL0
      (foreach T1 LL1
        (if (COMPAT-2L)
            (foreach T2 LL2
              (if (COMPAT-3L)
                  (setq TERCETS-V*V (cons (list T0 T1 T2) TERCETS-V*V)))))))
    (if TERCETS-V*V
        (setq *T* (itoa (length TERCETS-V*V)) *TT* (append *TT* (list *T*))
              SUDOKUS-V*V (append SUDOKUS-V*V (list (reverse TERCETS-V*V)))))
    (prompt (strcat "\nEntre " (nth *K* *KK*) " línies compatibles amb la "
                    (itoa (setq *K* (1+ *K*))) "ª, " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª i " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª,\nhi ha " *T* " tríades "
                    "compatibles amb la VARIANT ESCOLLIDA.\n")))
```

```
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              TERCETS-V*V () SUDOKUS-V*V ())
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
        (foreach T0 TT0
          (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0))
          (mapcar '(lambda (T1 CC1)
                     (setq K -1 OK T)
                     (while (and OK (< (setq K (1+ K)) 9))
                            (if (wcmatch (nth K CC1) (nth K CC0))
                                (setq OK ())))
                     (if OK (progn
                              (setq CC1 (PERF-COLS CC1))
                              (mapcar '(lambda (T2 CC2)
                                         (setq OK T K -1)
                                         (while (and OK (< (setq K (1+ K)) 9))
                                                (if (or (wcmatch
                                                          (setq J (nth K CC2))
                                                          (nth K CC0))
                                                        (wcmatch J (nth K CC1)))
                                                    (setq OK ())))
                                         (if OK (setq SUDOKUS-V*V
                                                      (cons (SUD-NUM) SUDOKUS-V*V)
                                                      S-V*V (cons T S-V*V))))
                                      TT2 CCC2))))
                   TT1 CCC1))
        (setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
              KK (length S-V*V) S-V*V (list (cons P S-V*V)))
        (prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                                       (nth 1 *TT*) " tríades 4-5-6 i les "
                                       (nth 2 *TT*) " tríades 7-8-9,"
                        "\nde solucions compatibles amb la VARIANT ESCOLLIDA "
                        "n'hi ha " (itoa KK) "."))
        (textscr)
        (SEGUIR ())
        (graphscr))))
```

Com que abans ja hem deixat clar que l'ús de solucions normalitzades no va ser cap opció sobrevinguda sinó la contemplada des del començament, i que la versió que us acabem d'oferir de **PERFIL-V*V** i **COMPAT-PERFILS** només tenia el propòsit d'apreciar amb detall els avantatges d'aquella opció, perquè així podrem copsar la reducció que s'opera en el nombre i profunditat de les comparacions, l'autor us estalviarà xifres aclaparadores sobre la durada que hauria tingut un procés totalment absent de simplificacions (val a dir que també ho ha fet per estalviar-s'ho ell mateix): ja n'hi haurà prou a veure com de llargues poden arribar a ser aquestes esperes, per imaginar-se quines proporcions haurien adquirit sense cap mena de filtratge.

Si les files emmagatzemades a **LINIES-V*V** pertanyen a una solució normalitzada, la selecció que durà a terme **PERFIL-V*V** podrà ser més restrictiva que la practicada, perquè les files hauran de complir unes condicions que ja havíem justificat en el capítol 1 (*Inventariar les solucions: un cul-de-sac*) però que anirà bé de repetir:
- La 1ª fila ha de començar amb **"1"**.
- La 2ª fila pot començar amb valors entre **"2"** i **"8"**.
- La 3ª fila pot començar amb valors entre **"3"** i **"9"**.
- La 4ª fila pot començar amb valors entre **"2"** i **"4"**.
- La 5ª fila pot començar amb valors entre **"3"** i **"8"**.
- La 6ª fila pot començar amb valors entre **"4"** i **"9"**.
- La 7ª fila pot començar amb valors entre **"3"** i **"7"**.
- La 8ª fila pot començar amb valors entre **"4"** i **"8"**.
- La 9ª fila pot començar amb valors entre **"5"** i **"9"**.

Sabent això, no cal aplicar la comparació **(wcmatch E L)** a tots els elements de **L1** sinó només als dels trams corresponents. Veiem-ho a la segona meitat de la funció:

```
(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V)
  (if (not L1) (C:CARREGA-123456789))
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
```

```
  (repeat 3
    (setq J (- J 9) K (+ K 3))
    (repeat 3
      (setq J (+ J 3) K (- K 3) W ())
      (repeat 3
        (setq J (- J 3) K (1+ K))
        (repeat 3
          (setq J (1+ J) V (nth J (nth K **V*V**)))
          (if (atom V) (setq W (cons V W)))))
      (cond ((= J 2) (setq A W))
            ((= J 5) (setq B W))
            (T       (setq C W))))
    (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
  (setq L1C (reverse L1C) J -1)
  (repeat 9
    (setq J (1+ J) K -1 L ())
    (repeat 9 (setq K (1+ K) C (nth J (nth K **V*V**))
                    L (if (atom C) (cons C L) L)))
    (setq L2C (cons L L2C)))
  (setq L2C (reverse L2C) LINIES-V*V () K -1)
  (terpri)
  (repeat 9
    (setq LINIA-V*V () L "" K (1+ K))
    (mapcar '(lambda (E F G)
               (setq L (strcat L (if (or F G)
                                     (if (atom E)
                                         (itoa E)
                                         (progn
                                           (setq C "]")
                                           (foreach H (append F G)
                                             (setq C (strcat (itoa H) C)))
                                           (strcat "[~" C)))
                                     "#"))))
            (nth K **V*V**) (nth K L1C) L2C)
    (cond ((= K 0) (setq A "123456789" B "198765432"))
          ((= K 1) (setq A "213456789" B "897654321"))
          ((= K 2) (setq A "312456789" B "987654321"))
          ((= K 3) (setq A "213456789" B "498765321"))
          ((= K 4) (setq A "312456789" B "897654321"))
          ((= K 5) (setq A "412356789" B "987654321"))
          ((= K 6) (setq A "312456789" B "798654321"))
          ((= K 7) (setq A "412356789" B "897654321"))
          (T       (setq A "512346789" B "987654321")))
    (foreach E (reverse (member B (reverse (member A L1))))
      (if (wcmatch E L) (setq LINIA-V*V (cons E LINIA-V*V))))
    (setq LINIES-V*V (cons (reverse LINIA-V*V) LINIES-V*V)
          *K* (itoa (length LINIA-V*V)) *KK* (cons *K* *KK*))
    (terpri)
    (prompt (strcat "\nEntre 362880 línies, de compatibles amb la " (itoa (1+ K))
                    "ª de la V. E. NORM. n'hi ha " *K* "."))
  (setq LINIES-V*V (reverse LINIES-V*V) L1 () *KK* (reverse *KK*))
  (terpri))
```

Pel que fa a **COMPAT-PERFILS**, limitant les solucions compatibles amb ****V*V**** a les normalitzades, obtindrem guanys considerables en les dues fases operatives: en la determinació dels requadres 9×3 **TT0** (el primer), **TT1** (el segon) i **TT2** (el tercer), compatibles amb la V. E. NORMALITZADA, i en la determinació final de les solucions compatibles **SUDOKUS-V*V** (en la versió precedent ja hem vist com aquesta variable i algunes altres juguen diferents papers al llarg de la funció, i ara ens referirem al segon, en què representa la llista de les solucions normalitzades compatibles):

- La comparació de cada 1ª [4ª o 7ª] fila **T0** de **LINIES-V*V** amb cada 2ª [5ª o 8ª] fila **T1**, i d'aquestes dues amb la 3ª [6ª o 9ª] **T2**, per donar lloc a requadres normalitzats 9×3 **TT0**, **TT1** i **TT2** compatibles amb la VERSIÓ ESCOLLIDA NORMALITZADA (de moment, no necessàriament compatibles entre ells), només prosperarà si és **T0** < **T1** < **T2**.
- No comparem tots els primers requadres normalitzats 9×3 **(car SUDOKUS-V*V)** (ara ens referim al primer paper que juga **SUDOKUS-V*V**, com a conjunt de tres llistes compostes per tots els primers, segons i tercers requadres 9×3 compatibles) amb

tots els segons **(cadr SUDOKUS-V*V)** i, si prospera la comparació, cada parell de requadres candidat amb tots els tercers **(caddr SUDOKUS-V*V)**, sinó que únicament considerem la llista **(cadr SUDOKUS-V*V)** a partir del requadre en què la 4ª fila comença amb el més petit dels valors "2", "3" i "4", exclosos els inicials de la 2ª i 3ª fila (si cap 4ª fila comença amb un valor diferent dels inicials de la 2ª i 3ª, llavors aquest primer requadre és incompatible i passarem al següent), i únicament considerem la llista **(caddr SUDOKUS-V*V)** a partir del requadre en què la 7ª fila comença amb el més petit dels valors "3", "4", "5", "6" i "7", exclosos els inicials de la 2ª, 3ª, 4ª, 5ª i 6ª fila (si cap 7ª fila comença amb un valor diferent dels inicials de la 2ª, 3ª, 4ª, 5ª i 6ª, llavors aquest segon requadre és incompatible i passarem al següent). Encara podríem afinar més les comparacions i que, a més d'excloure de **(cadr SUDOKUS-V*V)** i **(caddr SUDOKUS-V*V)** la part de llista anterior a la primera fila d'atac (localitzada per la funció **LOCT**), exclogués els requadres 9×3 en què el valor inicial de la primera fila coincidís amb els de totes les altres files precedents, però el rebombori que hauríem d'armar per aconseguir això segurament no compensaria: millor que deixem a **wcmatch**, tot i que sigui més endavant, posar de manifest la incompatibilitat entre els requadres que hagin passat pel nostre garbell menys selectiu.

Potser convé aclarir que, a més d'escapçar la llista **(cadr SUDOKUS-V*V)** deixant **PRIM1** com a primer element, i la llista **(caddr SUDOKUS-V*V)** deixant **PRIM2** com a primer element, cal escapçar **CCC1** i **CCC2** homòlogament (ho farà la funció **MEMBRE**), per mantenir la coherència dels arguments subministrats al tàndem **mapcar** + **lambda**, que apareix a dos nivells per organitzar les comparacions. Quant a la funció **LOCT** que indica per on cal escapçar (amb la subordinada **ACTLOC**), l'aparent complicació es justifica per la conveniència d'evitar, en passar de cada requadre 9×3 de **TT1** o **TT2** al següent, la comparació dels caràcters inicials de les primeres files, fent

```
(if (= (substr (car (nth K TT)) 1 1) (substr (car (nth (1+ K) TT)) 1 1)) ...)
```

per decidir si passem de llarg o registrem el caràcter a la llista **LOCT1** o **LOCT2**. En comptes d'això, únicament ens molestarem a identificar aquest caràcter inicial **(substr (car (nth K TT)) 1 1)** en el primer requadre i cada vegada que hi hagi un canvi, anomenant **A** el caràcter numèric immediatament superior i saltant-nos tots els requadres amb el caràcter inicial repetit mitjançant l'expressió

```
(while (< (car (nth (setq K (1+ K)) TT)) A))
```

que no necessita recórrer a la funció **substr** sinó que compara directament el text primera fila amb **A**. Per entendre'ns, en comptes de decidir que passem de llarg si un caràcter inicial és "**3**" i el següent també és "**3**", ho farem si la primera fila del requadre següent és "**398765421**" < **A** (**A** = "**4**"), però no si és "**412356789**" > **A**. Veiem la funció **COMPAT-PERFILS** optimitzada, amb les noves subordinades i amb una nova versió de les funcions **COMPAT-2L** i **COMPAT-3L** en què adoptarem restriccions a la comparació de files (**(< T0 T1)** i **(< T1 T2)**) que permetran d'inicialitzar **K** a **1**, per tal de comparar únicament els 8 últims caràcters:

```
(defun COMPAT-2L ()
  (setq K 1 OK T)
  (while (and OK (< K 9))
         (setq K (1+ K) OK (/= (substr T0 K 1) (substr T1 K 1)))))

(defun COMPAT-3L (/ M)
  (setq K 1 OK T)
  (while (and OK (< K 9))
         (setq K (1+ K) OK (/= (substr T1 K 1) (substr T2 K 1) (substr T0 K 1))))
  (if OK
      (progn
        (setq K -2)
        (repeat 2 (if OK (progn
                           (setq M "" K (+ 3 K))
                           (foreach L (list T0 T1 T2)
                             (setq M (strcat M (substr L K 3))))
                           (foreach N TOT
                             (if (and OK (not (wcmatch M N))) (setq OK ()))))))))
  OK)

(defun ACTLOC () (setq LOC (cons (cons A K) LOC)))

(defun LOCT (TT / A Z LOC L1)
  (setq A (substr (caar TT) 1 1) Z (substr (car (last TT)) 1 1) K 0)
  (ACTLOC)
```

```
  (while (< A Z)
         (setq A (itoa (1+ (atoi A))))
         (while (< (car (nth (setq K (1+ K)) TT)) A))
         (setq A (substr (car (nth K TT)) 1 1))
         (ACTLOC))
  (reverse LOC))

(defun SUPR (E L)
  (append (reverse (cdr (member E (reverse L)))) (cdr (member E L))))

(defun MEMBRE (K L) (repeat K (setq L (cdr L))) L)

(defun COMPAT-PERFILS (/ L1 LL0 LL1 L2 LL2 3L M N TERCETS-V*V CC0 CC1 CC2 CCC1
                        CCC2 TT0 TT1 TT2 LOCT1 LOCT2 TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  (repeat 9 (setq K (1+ K) TOT (cons (strcat "*" (itoa K) "*") TOT)))
  (repeat 3
    (setq LL0 (car LINIES-V*V) LL1 (cadr LINIES-V*V) LL2 (caddr LINIES-V*V)
          LINIES-V*V (cdddr LINIES-V*V) TERCETS-V*V ())
    (foreach T0 LL0
      (foreach T1 LL1
        (if (and (< T0 T1) (COMPAT-2L))
            (foreach T2 LL2
              (if (and (< T1 T2) (COMPAT-3L))
                  (setq TERCETS-V*V (cons (list T0 T1 T2) TERCETS-V*V)))))))
    (if TERCETS-V*V
        (setq *T* (itoa (length TERCETS-V*V)) *TT* (append *TT* (list *T*))
              SUDOKUS-V*V (append SUDOKUS-V*V (list (reverse TERCETS-V*V)))))
    (prompt (strcat "\nEntre " (nth *K* *KK*) " línies compatibles amb la "
                    (itoa (setq *K* (1+ *K*))) "ª, " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª i " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª,\nhi ha " *T* " tríades "
                    "normalitzades compatibles amb la V. E. NORMALITZADA.\n")))
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
        (foreach T0 TT0
          (if (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0)
                    L1 (SUPR (substr (cadr T0) 1 1) TOT)
                    L1 (SUPR (substr (caddr T0) 1 1) L1)
                    PRIM1 (cdr (assoc (car L1) LOCT1)))
              (mapcar '(lambda (T1 CC1)
                         (setq L2 (cdr L1) K -1 OK T)
                         (while (and OK (< (setq K (1+ K)) 9))
                                (if (wcmatch (nth K CC1) (nth K CC0))
                                    (setq OK ())))
                         (if (and OK (setq CC1 (PERF-COLS CC1)
                                           L2 (SUPR (substr (cadr T1) 1 1) L2)
                                           L2 (SUPR (substr (caddr T1) 1 1) L2)
                                           PRIM2 (cdr (assoc (car L2) LOCT2))))
                             (mapcar '(lambda (T2 CC2)
                                        (setq OK T K -1)
                                        (while (and OK (< (setq K (1+ K)) 9))
                                               (if (or (wcmatch
                                                         (setq J (nth K CC2))
                                                         (nth K CC0))
                                                       (wcmatch J (nth K CC1)))
                                                   (setq OK ())))
                                        (if OK (setq SUDOKUS-V*V
                                                     (cons (SUD-NUM) SUDOKUS-V*V)
                                                     S-V*V (cons T S-V*V))))
                                     (member (nth PRIM2 TT2) TT2)
                                     (MEMBRE PRIM2 CCC2))))
                      (member (nth PRIM1 TT1) TT1) (MEMBRE PRIM1 CCC1))))
```

```
        (setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
              KK (length S-V*V) S-V*V (list (cons P S-V*V)))
        (prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                                       (nth 1 *TT*) " tríades 4-5-6 i les "
                                       (nth 2 *TT*) " tríades 7-8-9,"
                        "\nde solucions normalitzades compatibles amb la V. E. "
                        "NORMALITZADA n'hi ha " (itoa KK) "."))
        (textscr)
        (SEGUIR ())
        (graphscr))))
```

Com hem avançat en obrir la pàgina 380, la missió de la llista de subllistes **S-V*V** és portar el recompte de les **KK** solucions compatibles (normalitzades, finalment), des del moment en què treiem la primera **FOTO1** de la situació (inici de **PERFIL-V*V**) i se'ns informa d'aquest valor (en acabar **COMPAT-PERFILS**) fins a treure'n la segona, quan només en queda una. Començant per **P**, la casella que en ser activada dispara l'alarma **POST**, cada element de **S-V*V** és una llista encapçalada per les coordenades d'una nova casella i seguida per **KK** valors binaris **T** o **nil**, segons que l'activació sigui o no compatible amb cadascuna de les solucions normalitzades enregistrades a **SUDOKUS-V*V**: com és lògic, en la primera subllista (de fet l'última, perquè **S-V*V** es forma amb **cons** i no ens molestem a invertir-la amb **reverse**) tots els **KK** valors seran **T**. L'actualització de **S-V*V** a cada nou clic (activació o desactivació) va a càrrec de la funció **NUMCOMPAT**, bàsicament. Pel que fa a les dues possibles formes d'intervenir-hi, les teniu a la fi del fragment de **FES-PUBLIC** que hem seleccionat,

```
                  ..................................
                  (if POST
                      (if V ; (CLIC) en una casella plena.
                          (if (member P FOTO1)
                              (progn .......)
                              (progn
                                (setq I (assoc P S-V*V)
                                      J (reverse(cdr (member I (reverse S-V*V))))
                                      S-V*V (append J (cdr (member I S-V*V)))
                                      I (caar S-V*V)
                                      J (nth (car I) (nth (cadr I) **9*9**)))
                                (NUMCOMPAT I J ())))
                          (NUMCOMPAT P N T))) ; (CLIC) en una casella buida.
                          ....................................................
```

i corresponen a un clic sobre una casella ja activada però que no figurava a **FOTO1** (és a dir, emplenada després que **COMPAT-PERFILS** fés l'inventari **SUDOKUS-V*V** de les solucions normalitzades compatibles) o sobre alguna que encara estava desactivada; una tercera possibilitat seria fer clic sobre una casella activada i que ja sortís a **FOTO1**, cas en què no hi hauria més remei que repetir **PERFIL-V*V** i **COMPAT-PERFILS** per treure una nova **FOTO1** i tornar a formar **SUDOKUS-V*V**. Veureu que el primer dels accessos esmentats a **NUMCOMPAT** està precedit d'una assignació múltiple, que té per objecte esborrar de **S-V*V** la subllista corresponent a la casella que desactivem i recuperar les coordenades **I** i el valor d'ocupació **J** de l'última casella activada i no revocada, que són utilitzats com arguments, en contraposició al segon accés, en què les coordenades **P** i el valor assignat **V** corresponen als de la casella sobre la qual hem fet clic. Però encara hi ha un tercer argument, que en el primer cas val **nil** i en el segon **T**, així que ja és hora que oferim el codi de **NUMCOMPAT**, tercera i última de les funcions que ens havíem proposat d'explicar en el present capítol:

```
(defun NUMCOMPAT (P N NOU / PS)
  (setq K 0 KK 0)
  (foreach SUD SUDOKUS-V*V
    (setq K (1+ K)
          OK (if NOU (= (nth (car P) (nth (cadr P) SUD)) N) T)
          PS (if NOU (cons OK PS)))
    (if OK (progn
             (foreach S S-V*V (if (and OK (not (nth K S))) (setq OK ())))
             (if OK (setq KK (1+ KK))))))
  (if NOU (setq S-V*V (cons (cons P (reverse PS)) S-V*V))))
```

Veureu que aquest tercer argument, anomenat **NOU**, és allò que regula el trànsit per la funció, segons que sigui accedida després d'una desactivació (**NOU** = **nil**, cas en què la supressió de la subllista obsoleta de **S-V*V** és externa i prèvia) o després

d'una activació (**NOU** = **T**, cas en què caldrà actualitzar **S-V*V**, aportant-hi la nova subllista). En aquest segon cas, per apreciar la compatibilitat de la part púbica del sudoku (incloent la nova casella) amb cadascuna de les solucions normalitzades **SUDOKUS-V*V**, aplicarem successivament 2 condicions: que **N** coincideixi amb el valor de la casella homòloga **P** de la solució (la coincidència quedarà reflectida en la nova subllista de **S-V*V**) i que totes les caselles activades també hagin satisfet aquesta condició (ho sabrem analitzant les subllistes precedents), i el comptador **KK** només s'incrementarà (recollint la solució) en complir-se ambdues condicions. Per contra, quan **NOU** = **nil**, **NUMCOMPAT** dóna per complerta la primera condició i es limita a verificar la segona. En aquest sentit, els valors **I** i **J** transmesos a **P** i **N** podien haver estat arbitraris, i si ens hem molestat a calcular **(caar S-V*V)** i **(nth (car I) (nth (cadr (caar S-V*V)) **9*9**))** serà probablement pel rastre d'una versió anterior, en que l'herència negativa de les subllistes **S-V*V** precedents es transmetia a cada nova subllista, cosa que evitava tenir que repassar-les totes, cada vegada, per calcular **KK** (cada nova subllista de **S-V*V** tenia menys membres **T**, i l'activació que assolia el desitjat **KK** = **1** hi aportava una subllista amb només un membre **T**); però aquesta versió fallava així que volíem actualitzar **S-V*V** davant d'una desactivació, perquè en una llista on cada element estava condicionat pels precedents no n'hi havia prou a eliminar el corresponent a la casella desactivada.

Amb les precisions de l'últim capítol sobre **SUD-MIN**, i amb les del present sobre **PERFIL-V*V**, **COMPAT-PERFILS** i **NUMCOMPAT** (sols caldria afegir que la funció **FOTO-V*V** treu una instantània de l'escaquer 9×9 en el moment de l'activació de **POST**, creant una llista amb les coordenadas de les caselles emplenades), ja estem en condicions de completar la panoràmica sobre **FES-PUBLIC** que havíem començat en el penúltim, en prolongar-la a l'estudi de les iteracions **while** fins més enllà de l'accés **(CLIC)**. Podem distingir quatre etapes:

1) Comencem amb **POST** = **nil**. A cada casella que activem, **SUD-MIN** comença activant **POST** però immediatament miràrà si hem traspassat el llindar a partir del qual la part pública de la V. E. NORMALITZADA pot esdevenir un SUDOKU-PROBLEMA; si no és així, **POST** torna a desactivar-se.

2) Quan, finalment, **SUD-MIN** detecta que hem traspassat el llindar, **POST** es queda activat i s'executa el tàndem **PERFIL-V*V** + **COMPAT-PERFILS**, desencadenant les accions que ja coneixem:
- Obtenció de la llista **FOTO1** amb les posicions de les caselles activades en el moment de travessar el llindar.
- Obtenció de la llista **SUDOKUS-V*V** de solucions normalitzades compatibles amb aquesta situació, de longitud **KK**.
- Inicialització de la llista de subllistes **S-V*V**.
- Informació detallada de files i tríades de files compatibles, culminant amb el nombre **KK** de solucions compatibles.

3) Des d'aquest moment, seguirem activant caselles, **NUMCOMPAT** anirà actualitzant **S-V*V** i **KK**, i se'ns informarà puntualment d'aquest valor. Si **KK** > **1** (queda més d'una solució normalitzada compatible amb la part pública) no es produeix cap canvi i hem de seguir activant caselles. Si **KK** = **1** (només en queda una), podrem plantar-nos o seguir (afegint redundància al SUDOKU-PROBLEMA), però abans tenen lloc les accions següents:
- Actualització de la llista **FOTO1** amb les posicions de les caselles activades fins aquest moment.
- Activació de la variable binària **FOTO2**.
- Activació de la variable binària **KKK**.

4) Si hem decidit seguir, podem anar activant caselles sense haver-nos de sotmetre a l'examen de **SUD-MIN** i sense més limitació que la obvietat de tenir-les totes activades; només passarem per **NUMCOMPAT** (tot i saber d'entrada que **KK** = **1**, cal enregistrar aquestes caselles en **S-V*V**, en previsió de futures desactivacions).

Fins aquí el procés sembla d'allò més simple; però, ¿què passaria fent clic en una casella ja activada? Tot depèn de l'etapa on es produeixi la desactivació, a banda de l'actualització de **V*V** i **V**V** operada en **CLIC**:
- Si es produeix entre **1** i **2**, cap particularitat.
- Si es produeix entre **2** i **3**, dependrà de si la casella **P** que volem desactivar ja estava activada en fer inventari de les solucions normalitzades compatibles (és a dir, si sortia en la primera **FOTO1**) o ho va ser més tard. En aquest segon cas **NUMCOMPAT** determinarà el nombre **KK** de solucions normalitzades compatibles i ens

n'informarà. Però en el primer caldrà anar a una nova execució de **PERFIL-V*V** i **COMPAT-PERFILS** si, malgrat un advertiment al respecte, l'usuari persisteix en el seu propòsit. En aquest sentit, cal advertir que l'execució de les dues funcions no està condicionada a la sentència previa de **SUD-MIN**, com la primera vegada, de manera que un hipotètic retrocés a un estat d'activació anterior a aquest moment desencadenaria execucions **PERFIL-V*V** + **COMPAT-PERFILS** encara més llargues, sense cap necessitat objectiva de fer-les.
- Si es produeix entre **3** i **4**, dependrà de si la casella **P** que volem desactivar ja estava activada en haver arribat a una única solució normalitzada compatible (és a dir, si sortia a l'última **FOTO1**) o va ser més tard. En aquest segon cas només passarem per **NUMCOMPAT** (tot i saber que **KK** = **1**, s'ha de mantenir actualitzada la llista **S-V*V**) i se'ns informarà que segueix havent-hi una única solució. Però en el primer cal tornar a executar **PERFIL-V*V** i **COMPAT-PERFILS** si, malgrat un doble advertiment al respecte (a més d'avisar-nos que l'espera pot ser llarga, esmenta la possibilitat que un futur anunci de solució única amagués força redundància, per efecte de les activacions innecessàries d'última hora), l'usuari persisteix en el seu propòsit. Com abans, l'execució no està condicionada al dictamen previ de **SUD-MIN**.

POST és el principal regulador del trànsit, amb tres condicions successives (i per tant, no excloents mútuament) en el codi:
- **POST** = **T** recull totes les situacions que disposen de **FOTO1**: si són activacions o si són desactivacions no compreses en **FOTO1**, s'executa **NUMCOMPAT** i segueix sent **POST** = **T**; si la casella desactivada figurava a **FOTO1**, es demana una confirmació i **POST** passa a **nil** en cas positiu.
- **POST** = **nil** ha de recollir totes les situacions poden dur a l'execució de **PERFIL-V*V** i **COMPAT-PERFILS** (prèvia commutació a **POST** = **T**): les activacions provinents de **1** i les desactivacions en què **P** pertany a **FOTO1**, provinents de **2** o **3**, i ja confirmades (de l'última etapa, les que passen a **POST** = **nil**).
- **POST** = **T** recull situacions provinents de la penúltima i última etapa (totes les que ja disposen de **FOTO1**), per indicar-nos si hem de seguir activant caselles o ja ho podem deixar.

En aquesta dinàmica, a més del regulador **POST**, el valor de **KK** i la pertinença o no d'una casella **P** desactivada a l'última **FOTO1**, també tenen un paper important dues variables que, malgrat que en el pròxim capítol representaran altres continguts, aquí només intervenen com a modestos senyals binaris: **KKK**, que s'activa al final dels cicles oberts per l'execució de **PERFIL-V*V** i **COMPAT-PERFILS**, quan ja **KK** = **1** (i gràcies al qual evitem una nova **FOTO1** després de cada activació redundant), i **FOTO2**, que separa l'abans del després de la primera arribada a una única solució normalitzada compatible (això es tradueix en l'aparició d'un advertiment simple o doble, a l'hora de prevenir desactivacions accidentals, de molestes conseqüències quant a les esperes). Com que ara esteu en condicions d'interpretar correctament **FES-PUBLIC**, serà oportú repetir el codi que us havíem ofert dos capítols enrera:

```
(defun FES-PUBLIC (/ KK KKK Q QX QY V GR L1 SUDOKUS-V*V S-V*V FOTO1 FOTO2 *K*
                     *KK*)
  (setq **V*V** (INI-COORDS) **V**V** **V*V**)
  (prompt "\n.\n.\n.")
  (graphscr)
  (command "CVPORT" 2 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""
           "CVPORT" 3 "CAPA" "D" "NORM-1" ""
           "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  (YIN-YANG)
  (while (not (or (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if (= (car GR) 5)
             (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
             (if (and (= (car GR) 3)
                      (> (getvar "CVPORT") 1)
                      (setq P (cadr GR) PX (car P) PY (cadr P))
                      (equal (list PX PY) '(4 4) 4.48)
                      (or (< (- PX (setq X (fix PX))) 0.48)
                          (< (- (setq X (fix (1+ PX))) PX) 0.48))
                      (or (< (- PY (setq Y (fix PY))) 0.48)
                          (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                      (PUNT))
                 (progn
                   (setq Q ())
```

```
(CLIC)
(if POST
    (if V ; (CLIC) en una casella plena.
        (if (member P FOTO1)
            (progn
              (command "ESPACIOP" "BORRA" "LT" ""
                       "DESHACER" "M"
                       "INSERT" "C" "0,0" "" "" "")
              (RETOL-2 "Anul-lant" "aquesta" "casella,"
                       "caldrà" "recalcular" "solucions"
                       "compatibles" "i tornar a" "esperar...")
              (if FOTO2
                  (progn
                    (prompt "\n.\n.")
                    (SEGUIR T)
                    (command "DESHACER" "R" "DESHACER" "M"
                             "INSERT" "C" "0,0" "" "" "")
                    (RETOL-2 "...i, si surt" "\"EL SUDOKU"
                             "JA TÉ UNA" "SOLUCIÓ" "ÚNICA\", mai"
                               "no sabreu" "si sobren" "caselles"
                               "plenes.")))
              (initget "Si No")
              (setq CONF (getkword (strcat "\n.\n.\nConfirmeu "
                                           "l'anul·lació "
                                           "(S/N)? <N>: "))
                    CONF (if CONF CONF "No"))
              (if (= CONF "Si")
                  (setq POST ())
                  (progn
                    (command "DESHACER" "R" "UY" "ESPACIOM")
                    (CLIC))))
            (progn
              (setq I (assoc P S-V*V)
                    J (reverse(cdr (member I (reverse S-V*V))))
                    S-V*V (append J (cdr (member I S-V*V)))
                    I (caar S-V*V)
                    J (nth (car I) (nth (cadr I) **9*9**)))
              (NUMCOMPAT I J ())))
        (NUMCOMPAT P N T))) ; (CLIC) en una casella buida.
(if (not POST)
    (progn
      (if (and V (member P FOTO1))
          (setq POST T KKK ())
          (if (not V) (SUD-MIN **V*V** ())))
      (if POST
          (progn
            (if V
                (command "DESHACER" "R")
                (command "ESPACIOP" "BORRA" "LT" ""))
            (command "DESHACER" "M"
                     "INSERT" "C" "0,0" "" "" "")
            (RETOL-2 "Determinar" "quantes" "solucions"
                     "normalitza-" "des són" "compatibles"
                     "pot trigar" "hores." "Espereu...")
            (PERFIL-V*V)
            (COMPAT-PERFILS)
            (command "DESHACER" "R" "UY" "ESPACIOM")))))
(if POST
    (if (> KK 1)
        (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                        "normalitzades heu d'omplir més "
                        "caselles:\n"))
        (progn
          (if (not KKK) (setq FOTO1 (FOTO-V*V) FOTO2 T KKK T))
          (prompt (strcat "\n.\nEL SUDOKU JA TÉ UNA SOLUCIÓ "
                          "NORMALITZADA ÚNICA.\nOmplint més "
                          "caselles el fareu més fàcil ("
                          "<Intro> o < > per acabar)")))))))))
```

... però no hi eren totes les que ho són

Als bons lectors (és a dir, als lectors meticulosos i amb prou paciència com per haver seguit les explicacions) els haurà estranyat no haver llegit cap comentari a propòsit de la 5ª línia (començant pel final) del codi precedent. Fins i tot pot ser que alguns s'hagin molestat a buscar-ne l'explicació i hagin deduït que

(**if (not KKK) (setq FOTO1 (FOTO-V*V) FOTO2 T KKK T)**)

no pot ser altra cosa que la manera de permetre, si l'usuari ha decidit emplenar més caselles un cop assolida la solució normalitzada única, que pugui també fer-se enrera d'algun d'aquests afegitons, però amb la garantia que la casella a esborrar sigui de les prescindibles i no de les que figuraven a l'escaquer 9×9 en el moment de la proclamació. Perquè no només les caselles que figuraven a la **FOTO1** treta tot just ultrapassat el llindar **(SUD-MIN **V*V** ())** han d'estar entre les intocables (més ben dit, tocables però a condició de desencadenar una altra vegada el procés **(PERFIL-V*V)** + **(COMPAT-PERFILS)**): també les emplenades després i fins al moment de la solució normalitzada única, moment que caldrà immortalitzar amb una nova **FOTO1**.

Si l'autor hi ha passat de puntetes és perquè, quan ha optat per incloure aquí el codi esmentat ja sabia que aquesta versió de **FES-PUBLIC** no estava completa, però malgrat tot ho ha fet per centrar-se només en un vessant del problema i abordar la resta amb una part dels deures ja enllestits. De fet, va ser precisament en posar a punt (i a prova) aquesta funció, usant de conillet d'Índies el sudoku "difícil" de LA VANGUARDIA del 25-02-08, ja esmentat quan feia història de la gestació de **SUD-MIN**, que va aixecar la llebre: en deixar per al final la introducció del valor **4** a la casella central va comprovar que, quan sols quedava omplir aquesta casella, ja apareixia el missatge "EL SUDOKU JA TÉ UNA SOLUCIÓ NORMALITZADA ÚNICA". Es va estranyar força que, allò que durant un temps va creure que era representatiu d'un "sudoku mínim", amb 23 caselles plenes, ara posés de manifest la seva condició de sudoku redundant que es podia seguir despullant i deixar amb només 22. Però quan es va posar a resoldre'l manualment sense la casella central, va veure que no: amb 22 era un sudoku incomplet, amb més d'una solució. Per no fer-ho tan ensopit per al lector, reproduïm aquí el sudoku de LA VANGUARDIA, base de la VARIANT ESCOLLIDA (esquerra), el de la V. E. NORMALITZADA (centre) i la solució d'aquesta (dreta):

6 0 0	0 0 9	0 1 2
0 0 0	0 0 0	4 5 0
0 1 0	3 0 0	0 0 0
0 7 0	0 0 5	0 0 0
0 0 9	1 **4** 7	5 0 0
0 0 0	6 0 0	0 4 0
0 0 0	0 0 4	0 6 0
0 0 1	0 0 0	0 0 0
8 0 0	7 0 0	0 0 3

0 0 0	0 0 0	4 5 0
6 0 0	0 0 9	0 1 2
0 1 0	3 0 0	0 0 0
8 0 0	7 0 0	0 0 3
0 0 0	0 0 4	0 6 0
0 0 1	0 0 0	0 0 0
0 7 0	0 0 5	0 0 0
0 0 9	1 **4** 7	5 0 0
0 0 0	0 0 0	0 4 0

9 3 2	8 6 1	4 5 7
6 8 7	4 5 9	3 1 2
5 1 4	3 7 2	6 9 8
8 4 5	7 9 6	1 2 3
7 9 3	2 1 4	8 6 5
2 6 1	5 3 8	9 7 4
4 7 6	9 8 5	2 3 1
3 2 9	1 4 7	5 8 6
1 5 8	6 2 3	7 4 9

No va trigar massa a adonar-se que en el seu discurs havia estat arrossegant una ambigüitat que ara calia resoldre: no reflectia la mateixa situació dir "EL SUDOKU JA TÉ UNA SOLUCIÓ NORMALITZADA ÚNICA" que "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA"; la primera només era el preludi de la segona en el camí adoptat. En l'exemple, si ara considerem el sudoku del centre però desprovist de l'element 5,2 (**4**), a més de la solució reproduïda n'hi haurà 592 de no normalitzades però igualment vàlides i a tenir en compte. Emplenant una casella més amb el valor predeterminat, el nombre de solucions compatibles es reduirà en diferent mesura, com podeu veure aquí a la dreta, on a cada casella lliure del sudoku hi figura el nombre de solucions que quedarien si l'omplíssim: veieu que a més de l'esmentada n'hi ha una altra, la 1,3, que assoliria la SOLUCIÓ ÚNICA només omplir-la (també amb un **4**), i que emplenar les caselles 9,4, 9,5 o 7,8 no serveix de res. A títol de mostra, a la pàgina següent hi ha les 3 solucions no normalitzades que quedarien omplint 5,8.

187 221 188	422 176 220	--- --- 247
--- 291 027	024 004 ---	**593** --- ---
217 --- 034	--- 027 372	430 100 069
--- 291 042	--- 002 152	148 202 ---
238 212 204	211 148 ---	086 --- **593**
186 224 ---	024 067 027	020 478 **593**
001 --- 355	002 047 ---	197 201 204
008 067 ---	--- **001** ---	--- 038 355
204 247 058	--- 043 090	449 --- 055

```
9 3 2   8 6 1   4 5 7       3 9 2   8 6 1   4 5 7       2 9 3   8 6 1   4 5 7
6 8 7   4 5 9   3 1 2       6 8 7   4 5 9   3 1 2       6 8 7   4 5 9   3 1 2
5 1 4   3 7 2   6 9 8       5 1 4   3 7 2   6 8 9       4 1 5   3 7 2   6 8 9

8 4 5   7 9 6   1 2 3       8 4 5   7 2 6   1 9 3       8 5 4   7 2 6   1 9 3
7 9 3   2 1 4   8 6 5       7 2 3   9 1 4   8 6 5       7 3 2   9 1 4   8 6 5
2 6 1   5 3 8   9 7 4       9 6 1   5 3 8   2 7 4       9 6 1   5 3 8   2 7 4

3 7 6   9 4 5   2 8 1       1 7 6   2 4 5   9 3 8       1 7 6   2 4 5   9 3 8
4 2 9   1 8 7   5 3 6       4 3 9   1 8 7   5 2 6       3 4 9   1 8 7   5 2 6
1 5 8   6 2 3   7 4 9       2 5 8   6 9 3   7 4 1       5 2 8   6 9 3   7 4 1
```

Sortosament va descobrir que la feina feta no sols era aprofitable sinó ben útil, en la mesura que, com Colom (no l'àlies de l'autor sinó en Cristòfor el navegant), l'havia encertada en creure que la ruta occidental era més curta que l'oriental, i només errava en suposar-la més curta del compte: en el cas d'en Joan Colom, jugar amb solucions normalitzades per bregar amb un conjunt força més reduït no era mala idea; l'error era pensar que tot s'hauria acabat quan ja només en quedés una, però només adonar-se'n de seguida va posar fil a l'agulla i no ho va deixar fins que la versió ampliada de **FES-PUBLIC** no va abastar el nou horitzó. Ara, per a no fer més feixuga la progressió del text explicatiu, anirem directament a justificar els dispositius adoptats, partint del supòsit de disponibilitat d'aquesta nova versió.

Quan, per l'execució reiterada de **SUD-MIN**, s'activa finalment **POST** i es posa en marxa **COMPAT-PERFILS**, que conclou anunciant-nos "Entre les TT1 tríades 1-2-3, les TT2 tríades 4-5-6 i les TT3 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORM. n'hi ha SSNNCC", hem de fixar-nos en el missatge que segueix i que es repeteix a cada nova casella que emplenem (o, més ben dit, desemmascarem): "Amb SSNNCC solucions normalitzades heu d'omplir més caselles". Tret que d'entrada sigui SSNNCC = 1, aquest valor va decreixent fins que rebem l'esperada notícia: "Teniu 1 solució normalitzada compatible amb V. E. NORMALITZADA". Però tot seguit, fora d'alguns casos en què seguirem llegint "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA. Omplint més caselles el fareu més fàcil (<Intro> o < > per acabar)", el que ens caurà al damunt com una gerra d'aigua freda serà aquest rètol: "Però pot haver-n'hi més, de no normalitzades però igualment compatibles, així que espereu...".

Tot plegat pot semblar una broma de mal gust, sobretot tenint en compte la llarga espera que haurà precedit l'aparició del primer d'aquests missatges. Però, si hem seguit les explicacions fins aquí, no en podíem esperar altra cosa: en el còmput a càrrec de **COMPAT-PERFILS** es treballa sobre la VARIANT ESCOLLIDA NORMALITZADA (és a dir, amb les files ordenades numéricament a cada requadre 9×3 i els tres requadres ordenats numèricament, atenent les primeres files) de la VARIANT ESCOLLIDA de la SUDOKU-SOLUCIÓ adoptada (entre les variants permeses, obtingudes per transposició, permutació i/o substitució dels elements originals), i només es busquen solucions compatibles que també estiguin normalitzades, a fi i efecte de reduir el nombre d'elements en joc i, en definitiva, de comparacions. Ara bé, quan surt el missatge "Teniu 1 solució normalitzada compatible amb V. E. NORMALITZADA" el més probable és que no haguem arribat al cap del carrer i que el sudoku destacat en pantalla tingui més d'una solució: el missatge únicament indica que, de les SSCC solucions compatibles (i en general serà SSCC > 1) només n'hi ha una que sigui normalitzada. D'una cosa en podem estar ben segurs: mai no podrà passar que, de tenir més d'una solució normalitzada compatible, en emplenar una nova casella passem a no tenir-ne cap ni una (de SNC) perquè, si més no, encara hi restarà la solució predeterminada (VARIANT ESCOLLIDA NORMALITZADA) que només coneix l'usuari omniscient. En resum, quan ens anuncien "Teniu 1 solució normalitzada compatible amb V. E. NORMALITZADA. Però..." i quan rebem la notícia "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA..." se'ns està parlant de la mateixa solució (la normalitzada subministrada per nosaltres), sols que en el moment d'aparèixer el primer avís encara hi poden haver altres solucions compatibles i quan surt el segon, amb més caselles plenes, ja no n'hi ha cap més.

L'autor es creu en el deure de fer un incís aquí per aclarir que, amb l'estratègia ideada inicialment i que havíem exposat en el primer capítol (crear 123456789.txt, l'arxiu de files que ha resultat ser l'únic supervivent realitzable i aprofitable; un conjunt d'arxius KOMPAT-..., cadascun dels quals inventariava els requadres 9×3 normalitzats que començaven amb una fila del primer; un conjunt d'arxius XSN-..., cada un dels quals aplegava les solucions normalitzades constituïdes per requadres 9×3 normalitzats compatibles pertanyents al segon), hauríem anat a raure al mateix

lloc: tot i que l'orientació que finalment apuntàvem hagués facilitat la tasca de cerca, en relació a la primitiva (en remetre a un XSN-<pF1>-<pF4>-<pF7> concret, determinat per les posicions a 123456789.txt de la 1ª, 4ª i 7ª files de la V. E. NORMALITZADA), l'acumulació ingent d'informació i el treball previ per obtenir-la tampoc no haurien justificat un desplegament logístic tan desmesurat i que, al cap i a la fi, només hauria servit per a la primera etapa; a partir del moment en què cal donar el salt a totes les solucions compatibles, ja no ens serviria de res... tret que l'inventari abastés totes les solucions i no únicament les normalitzades.

Però, tornant al decebedor descobriment, tampoc no cal amoïnar-se massa, perquè quan s'esdevingui aquest fet l'escaquer 9×9 ja estarà força més ple que quan es va posar en marxa **COMPAT-PERFILS**, tret que fóssim tan hàbils emplenant caselles que haguéssim aconseguit de retardar al màxim aquest moment i haguéssim rebut com a primer missatge "Teniu 1 solució normalitzada compatible amb V. E. NORMALITXADA" (si, just després de polsar una tecla, sortís "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA. Omplint més caselles el faràs més fàcil (<Intro> o < > per acabar)", ja no caldria patir més, però tenir tanta sort no és massa probable), supòsit en què la funció esmentada ja se l'hauria trobat prou ple i se n'hauria sortit sense gran esforç. Aquesta situació més favorable permetrà de construir la nova llista **SUDOKUS-V*V** de solucions compatibles ordinàries (de normalitzades només resta la predeterminada) amb un dispositiu recursiu similar al de **RE-MEMBER**, que com aquesta funció ha de disposar de les llistes ****9*9**** i **LLL** (feu memòria d'aquestes pseudomatrius 9×9: a la primera, les caselles plenes exhibien el valor i les buides la posició, just el paper que ara fa ****V*V****; a la segona, les ocupades eren nul·les i les buides estaven representades per una llista de 9 elements on només apareixien els valors compatibles i la resta eren nuls). Així com **PERFIL-V*V** havia començat treient una instantània de ****V*V**** (en realitat la llista **FOTO1** creada per **FOTO-V*V** registrava les posicions de les caselles plenes), ara la funció **ANORMALS** (el nom es refereix a les solucions no normalitzades) comença treient-ne una altra (**FOTO2**), organitza la pseudomatriu **LLL** i després, a partir de la primera casella buida localitzada a **(COMPLET-9*9 **V*V**)**, es llança a explorar el sudoku amb el dispositiu **RE+MEMBER**, per refer la llista **SUDOKUS-V*V** de les solucions compatibles amb la nova ****V*V**** (la de **FOTO2**) i, mitjançant **KKK** (que enregistra per a cada casella emplenada les solucions **SUDOKUS-V*V** que es mantenen compatibles i les que ja no, com fèiem amb les solucions normalitzades), la llista **S-V*V** que emmagatzema els **KKK** successius. La variable **KK** representa com abans el nombre de solucions compatibles (que ara no seran normalitzades, tret d'una) i **KKK**, a més de servir de *buffer* a **S-V*V**, juga un altre paper: en haver sorgit del no-res, marca la separació entre l'etapa iniciada a **COMPAT-PERFILS** (quan **SUDOKUS-V*V**, **S-V*V** i **KK** es referien a solucions compatibles normalitzades, amb el ****V*V**** reflectit a **FOTO1**) i la que comença amb **ANORMALS** (**SUDOKUS-V*V**, **S-V*V** i **KK** referits a solucions compatibles ordinàries, amb el nou ****V*V**** reflectit a **FOTO2**). Ara bé, com que l'aparició de **FOTO2**, substituint a **FOTO1**, ja és prou vàlida com a fita per delimitar aquests territoris, reservarem a **KKK** una altra missió. Altres variables que intervindran en la complexa regulació del trànsit són **V** i **POST**, assenyalant quan cal rellançar **COMPAT-PERFILS** (per haver anul·lat una casella registrada a **FOTO1**) o **ANORMALS** (per haver-ne anul·lat una de registrada a **FOTO2**), però abans d'abordar el tema del control del flux entre cada **(CLIC)** i **(CLIC)** ens ocuparem de **RE+MEMBER** i del paral·lelisme d'aquesta funció amb la quasi homònima **RE-MEMBER**.

Si en **RE-MEMBER** només calia trobar una solució derivada de l'assignació d'un valor a la casella lliure escollida, per poder-ne legitimar l'assignació d'aquest valor (havent optat per A, estàvem construint una solució i en cada pas es prenien dues decisions: quina casella emplenàvem i amb quin valor ho fèiem), en **RE+MEMBER** es tracta de localitzar totes les solucions compatibles amb una situació concreta. Fins i tot el nom pretén expressar aquesta diferència respecte a **RE-MEMBER**: ara no n'hi ha prou a saber si existeixen solucions compatibles amb les caselles plenes i amb l'assignació d'uns valors determinats (els admissibles en primera instància) a la que anem a omplir, sinó que volem sumar (inventariar) totes les compatibles amb l'estat reflectit a **FOTO2**. Tot i que l'usuari aquí juga sobre una solució concreta i a cada pas només exerceix un grau de llibertat (la posició o casella a emplenar, perquè el valor d'assignació ja està predeterminat), hagi optat per A, B o C, la necessitat o no de seguir-ne emplenant més depèn del nombre de solucions possibles en absència de condicionament, i en això ambdós plantejaments són força semblants.

Presentant de primer el context on funciona **RE+MEMBER**, **ANORMALS** es posarà en marxa quan, després d'emplenar la casella de posició **P**, **NUMCOMPAT** detecti que **KK** = **1**, i respondrà a l'esquema següent:

ANORMALS:
- Fes **SUDOKUS-V*V** = **KKK** = **nil**
- Crea **LLL**
- Avalua **RE+MEMBER**
- Completa **KKK** afegint-hi **P** com a primer element, i inicialitza **KK** i **S-V*V**

Pel que fa a **RE+MEMBER**, serà una funció recursiva definida així:

RE+MEMBER:
- Si l'escaquer 9×9 està complet, actualitza **SUDOKUS-V*V** (afegint-lo com solució) i **KKK** (afegint-hi **T**)
- Si no:
  - Cerca la primera casella buida
  - Amb cadascun dels valors que, en primera instància, admeti aquesta casella:
    - Emplena-la
    - Avalua **RE+MEMBER**

És a dir que, quan l'escaquer sigui ple, actualitzarem **SUDOKUS-V*V**, **KKK** i provarem altres valors en aquesta última casella buida (inútilment, perquè si un valor era compatible amb la resta de caselles -plenes amb valors concrets- cap més ja no ho podrà ser) i tornarem al nivell de recursió precedent (casella buida anterior) on, si encara quedaven valors per provar, provarem el següent i avançarem un nivell de recursió (passant a la primera casella lliure, i així successivament, fins haver emplenat l'escaquer o haver trobat pel camí alguna casella que no admeti cap valor en primera instància) o, si no en queden, tornarem al nivell (casella) precedent. La recursió s'acabarà quan haguem tornat a la primera casella buida i ja no quedin més valors per provar. L'objectiu del procés és l'obtenció de **SUDOKUS-V*V** i **KKK**.

Si, per evitar un ball de fulls, reproduïm aquí l'esquema de **RE-MEMBER** vist abans,

RE-MEMBER:
- Si **COMPLET** = **nil**:
  - Si l'escaquer 9×9 està complet, fes **COMPLET** = **T**
  - Si no:
    - Cerca la primera casella buida
    - Amb cadascun dels valors que, en primera instància, admeti aquesta casella:
      - Emplena-la
      - Avalua **RE-MEMBER**
- **COMPLET**

tindrem una formalització semblant a l'esquema **RE+MEMBER**. Però haurem fet trampa, perquè estarem ignorant la funció **foreach** (pertanyent a **MASCARA**) que l'emmarca, i l'emplenament de la casella activada (que pot no coincidir amb la primera casella buida, a l'inici de **RE-MEMBER** i que només s'executa la primera vegada). Serem més fidels al procés (tot i forçar una simplicitat que sacrifica alguns matisos) fent

RE-MEMBER ampliat:
- Amb cadascun dels valors que, en primera instància, admeti la casella activada:
  - Emplena-la
  - Fes **COMPLET** = **nil**
  - Avalua **RE-MEMBER**

en què **RE-MEMBER** respondria a l'esquema anterior. Ara podríem dir que el procés consisteix a provar en la casella activada tots els valors admissibles en primera instància, i amb cadascun d'ells executar una funció (**RE-MEMBER**) que s'assembla molt a **RE+MEMBER** tret que, quan l'escaquer sigui ple només farem **COMPLET** = **T** i això durà a remuntar tots els nivells recursius precedents, sense cap més acció: l'arribada del valor **T** al primer nivell donarà llum verda al candidat en qüestió. En tots els altres aspectes, la similitud entre **RE-MEMBER** i **RE+MEMBER** serà total: si en qualsevol nivell de recursió encara queden valors per provar, provarem el següent i avançarem un nivell de recursió (passant a la primera casella buida i així successivament, fins haver emplenat l'escaquer o haver trobat pel camí alguna casella que no admet cap valor en primera instància); si ja no en queden, tornarem al nivell precedent. La recursió s'acabarà quan haguem tornat a la primera casella buida i no quedin més valors per provar. L'objectiu del procés és esbrinar quins candidats a la casella han donat lloc a una solució (i l'usuari pot utilitzar-los) o a cap: únicament en aquest cas, partint d'una casella i d'un valor, l'exploració amb **RE-MEMBER** seguirà el mateix itinerari que si s'hagués efectuat amb **RE+MEMBER**.

Però ja que ens hem molestat a baixar un esquema des del capítol *3- Etapa prèvia: crear una solució, I (embolica, que fa fort!)*, no costa gens fer el mateix amb el codi implicat, que situem entre aquest capítol i el que segueix, *4- Etapa prèvia: crear una solució, II (la difícil simplicitat)*. I, precedint la funció **RE-MEMBER**, veieu també els fragments de **MASCARA** (en dèiem **RE-MEMBER ampliat**) que l'emmarquen:

```
  ..............
  (foreach N L1A9
    ..................
    (if (and (member N L)
             ............
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn ...) ...))
    ..................

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R))))))))
  OK)
```

En resum, per apreciar-ne similituds i diferències,
1) En tots dos casos l'exploració comença per la primera casella buida (assajant els valors 1... 9 admissibles en primera instància) i segueix ordenadament per les demés.
2) A **RE-MEMBER** es partia d'una excepció **INI**cial en la qual, a les caselles plenes efectivament s'hi afegia la que anàvem a emplenar (en cada sèrie de recursions preníem un dels valors 1... 9 admissibles en primera instància). En **RE+MEMBER**, en canvi, la situació inicial es limita a les caselles plenes efectivament.
3) A **RE-MEMBER** la variable **COMPLET** (acabàvem anomenant-la **OK**) s'utilitzava per interrompre l'exploració en trobar la primera solució (si no n'hi havia cap, en quedar la primera casella buida sense valors compatibles en primera instància. A **RE+MEMBER**, en canvi, cal trobar totes les solucions i per això en prescindim.

El parentiu indubtable entre **RE-MEMBER** i **RE+MEMBER** (malgrat haver decidit, en la primera, traslladar-hi la inicialització de **R-9*9** i **R-LLL**, sota el control de **INI**) es dilueix en considerar les unitats operatives **RE-MEMBER** < **MASCARA** < **OMPLE-SUDOKU** d'una banda i **RE+MEMBER** < **ANORMALS** de l'altra, raó per la qual trobem interessant presentar abans de seguir la traducció a codi de les dues últimes funcions, tot i que al final d'aquest mateix capítol ens veurem empesos a introduir-hi canvis per millorar-ne el rendiment:

```
(defun RE+MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if (COMPLET-9*9 R-9*9)
      (setq SUDOKUS-V*V (cons R-9*9 SUDOKUS-V*V)
            KKK (cons T KKK))
      (progn
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
        (foreach E L1A9
          (if (member E R-N)
              (progn
                (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                (RE+MEMBER (car 2R) (cadr 2R))))))))

(defun ANORMALS (/ L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P)
  (setq L0A8 '(0 1 2 3 4 5 6 7 8)
        FOTO1 () FOTO2 (FOTO-V*V)
        KKK () SUDOKUS-V*V ())
  (foreach X L0A8
    (setq XX ())
    (foreach Y L0A8
      (if (atom (setq V (ELEMENT **V*V**)))
          (setq XX (cons V XX))))
    (setq XXX (cons XX XXX)))
  (setq XXX (reverse XXX))
```

```
  (foreach Y L0A8
    (setq YY ())
    (foreach X L0A8
      (if (atom (setq V (ELEMENT **V*V**)))
          (setq YY (cons V YY))))
    (setq YYY (cons YY YYY)))
  (setq YYY (reverse YYY))
  (foreach YY '(0 3 6)
    (foreach XX '(0 3 6)
      (setq L () Y -1)
      (foreach VV **V*V**
        (setq Y (1+ Y) X -1)
        (if (not (or (< Y YY) (> Y (+ YY 2))))
            (progn
              (foreach V VV
                (setq X (1+ X))
                (if (not (or (< X XX) (> X (+ XX 2))))
                    (if (atom V) (setq L (cons V L))))))))
      (setq XY (cons L XY))))
  (setq XY (reverse XY))
  (foreach Y L0A8
    (setq LL ())
    (foreach X L0A8
      (setq L ())
      (if (listp (ELEMENT **V*V**))
          (foreach E L1A9
            (setq L (cons (if (or (member E (nth X XXX)) (member E (nth Y YYY))
                                  (member E (nth (+ (/ X 3) (* (/ Y 3) 3)) XY)))
                              () E)
                          L))))
      (setq LL (cons (reverse L) LL)))
    (setq LL (reverse LL) LLL (cons LL LLL)))
  (setq LLL (reverse LLL))
  (RE+MEMBER **V*V** LLL)
  (setq KK (length KKK) KKK (cons P KKK) S-V*V (list KKK)))
```

Vist **RE+MEMBER**, ja és hora d'ocupar-se del control del flux entre **(CLIC)** i **(CLIC)**. Com en l'etapa anterior (d'ara endavant l'anomenarem etapa "FOTO 1-2", des que es posa en marxa **COMPAT-PERFILS** fins a quedar-nos amb una única solució normalitzada compatible i es posa en marxa **ANORMALS**), en l'actual (que anomenarem etapa "FOTO 2-2", des d'aquest esdeveniment fins a quedar-nos amb una sola solució compatible, que per definició serà la normalitzada) gairebé totes les complicacions deriven de la possibilitat de picar sobre caselles emplenades per tornar a deixar-les buides. Si completem aquesta cronologia amb l'etapa prèvia a "FOTO 1-2" (que anomenarem etapa "PRE-FOTO") i la posterior a "FOTO 2-2" (que anomenarem etapa "POST-FOTO"), quan la revocació afecti una casella emplenada en el transcurs de la mateixa etapa no haurem d'efectuar cap operació especial, fora d'actualitzar **S-V*V** i **KK**, però quan esborrem una casella emplenada per darrera vegada en alguna de les etapes precedents, haurem de rebre un advertiment sobre les conseqüències que pot tenir això (una altra llarga espera i/o, si el resultat és un anunci de solució única, la impossibilitat de saber si el sudoku és o no redundant en relació a la solució) i tenir l'oportunitat de confirmar l'acció o fer-nos enrera. I quan, malgrat tot, insistim, s'haurà de produir la inicialització de **FOTO1** o **FOTO2** i de **SUDOKUS-V*V**. Aquesta eventualitat (de què l'etapa "PRE-FOTO" en queda al marge, per definició) i les conseqüències que se'n deriven en cada cas queden reflectides en l'esquema:

Etapa "PRE-FOTO":
- Es caracteritza perquè:
  - **POST** = **nil**.
  - A cada **(CLIC)**:
    - S'actualitzen ****V*V**** i ****V**V****.
    - Si el **(CLIC)** emplena una casella cal passar el control **SUD-MIN**.
- Omplir caselles és obligat.
- Buidar caselles mai no té conseqüències transcendents.
- S'acaba quan, després del control **SUD-MIN**, **POST** = **T**:
  - S'executa el tàndem **PERFIL-V*V** i **COMPAT-PERFILS**, que:
    - Crea **FOTO1** i **SUDOKUS-V*V** (solucions normalitzades compatibles).
    - Inicialitza **S-V*V** i **KK** (nombre de solucions normalitzades compatibles).

Etapa "FOTO 1-2":
- Es caracteritza perquè:
  - **POST** = **T**, **FOTO1** = **T** i **KK** > **1**.
  - A cada **(CLIC)**:
    - S'actualitzen ****V*V**** i ****V**V****.
    - Si no afecta caselles registrades a **FOTO1**:
      - S'actualitzen **S-V*V** i **KK**.
- Omplir caselles és obligat.
- Buidar caselles:
  - Si no estan registrades a **FOTO1** no té conseqüències transcendents.
  - Si hi estan registrades:
    - S'adverteix l'usuari sobre les conseqüències de validar el **(CLIC)** i se li dóna opció a fer-se enrera:
      - Si decideix fer-se enrera:
        - Un nou **(CLIC)** revoca el precedent i restableix el valor de la casella.
      - Si decideix tirar endavant i buidar la casella:
        - S'executa el tàndem **PERFIL-V*V** i **COMPAT-PERFILS**, que:
          - Actualitza **FOTO1** i **SUDOKUS-V*V** (solucions normalitzades compatibles).
          - Inicialitza **S-V*V** i **KK** (nombre solucions normalitzades compatibles).
        - Tornem a l'inici de l'etapa.
- S'acaba quan **KK** = **1**:
  - S'executa **ANORMALS**, que:
    - Crea **FOTO2** i actualitza **SUDOKUS-V*V** (solucions compatibles).
    - Inicialitza **S-V*V** i **KK** (nombre de solucions compatibles).

Etapa "FOTO 2-2":
- Es caracteritza perquè:
  - **POST** = **T**, **FOTO2** = **T** i **KK** > **1**.
  - A cada **(CLIC)**:
    - S'actualitzen ****V*V**** i ****V**V****.
    - Si no afecta caselles registrades a **FOTO2**:
      - S'actualitzen **S-V*V** i **KK**.
- Omplir caselles és obligat.
- Buidar caselles:
  - Si no estan registrades a **FOTO2** no té conseqüències transcendents.
  - Si hi estan registrades:
    - S'adverteix l'usuari sobre les conseqüències de validar el **(CLIC)** i se li dóna opció a fer-se enrera:
      - Si decideix fer-se enrera:
        - Un nou **(CLIC)** revoca el precedent i restableix el valor de la casella.
      - Si decideix tirar endavant i buidar la casella:
        - S'executa **ANORMALS**, que:
          - Actualitza **FOTO2** i **SUDOKUS-V*V** (solucions compatibles).
          - Inicialitza **S-V*V** i **KK** (nombre de solucions compatibles).
        - Tornem a l'inici de l'etapa.
- S'acaba quan **KK** = **1**:
  - S'executa **ANORMALS**, que:
    - Actualitza **FOTO2** (no cal **SUDOKUS-V*V**, **S-V*V** ni **KK**, perquè l'unica solució compatible és la solució normalitzada ****9*9****).

Etapa "POST-FOTO":
- Es caracteritza perquè:
  - **POST** = **T**, **FOTO2** = **T** i **KK** = **1**.
  - A cada **(CLIC)**:
    - S'actualitzen ****V*V**** i ****V**V****.
- Omplir caselles és opcional (per introduir redundància en el sudoku).
- Buidar caselles:
  - Si no estan registrades a **FOTO2** no té conseqüències transcendents.
  - Si hi estan registrades:
    - S'adverteix l'usuari sobre les conseqüències de validar el **(CLIC)** i se li dóna opció a fer-se enrera:
      - Si decideix fer-se enrera:
        - Un nou **(CLIC)** revoca el precedent i restableix el valor de la casella.
      - Si decideix tirar endavant i buidar la casella:
        - S'executa **ANORMALS**, que:
          - Actualitza **FOTO2** i crea **SUDOKUS-V*V** (solucions compatibles).
          - Inicialitza **S-V*V** i **KK** (nombre de solucions compatibles).
        - Tornem a l'inici de l'etapa "FOTO 2-2".
- S'acaba quan l'usuari polsa <CR> o <espai>.

En realitat l'última etapa es podria refondre amb la precedent, que quedaria així:

Etapa "FOTO 2-2":
- Es caracteritza perquè:
  - **POST** = **T** i **FOTO2** = **T**.
  - A cada **(CLIC)**:
    - S'actualitzen ****V*V**** i ****V**V****.
    - Si no afecta caselles registrades a **FOTO2**:
      - Si **KK** > **1**:
        - S'actualitzen **S-V*V** i **KK**.
- Omplir caselles:
  - És obligat, quan **KK** > **1**.
  - És opcional (per introduir redundància en el sudoku), quan **KK** = **1**.
- Buidar caselles:
  - Si no estan registrades a **FOTO2** no té conseqüències transcendents.
  - Si hi estan registrades:
    - S'adverteix l'usuari sobre les conseqüències de validar el **(CLIC)** i se li dóna opció a fer-se enrera:
      - Si decideix fer-se enrera:
        - Un nou **(CLIC)** revoca el precedent i restableix el valor de la casella.
      - Si decideix tirar endavant i buidar la casella:
        - S'executa **ANORMALS**, que:
          - Actualitza **FOTO2** i **SUDOKUS-V*V** (solucions compatibles).
          - Inicialitza **S-V*V** i **KK** (nombre de solucions compatibles).
        - Tornem a l'inici de l'etapa.
- La primera vegada que **KK** = **1**:
  - S'executa **ANORMALS**, que:
    - Actualitza **FOTO2** (no cal **SUDOKUS-V*V**, **S-V*V** ni **KK**, perquè l'unica solució compatible és la solució normalitzada ****9*9****).
- S'acaba quan l'usuari polsa <CR> o <espai>.

Amb totes aquestes consideracions, la funció iterativa **FES-PUBLIC** s'haurà ampliat fins a quedar en la forma:

```
(defun FES-PUBLIC (/ KK KKK Q QX QY NV V GR L1 SUDOKUS-V*V S-V*V FOTO1 FOTO2 *K*
                     *KK*)
  (setq **V*V** (INI-COORDS) **V**V** **V*V**)
  (prompt "\n.\n.\n.")
  (graphscr)
  (command "ESPACIOM" "CVPORT" 2 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""
           "CVPORT" 3 "CAPA" "D" "NORM-1" ""
           "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  (YIN-YANG)
  (while (not (or (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if (= (car GR) 5)
             (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
             (if (and (= (car GR) 3)
                      (> (getvar "CVPORT") 1)
                      (setq P (cadr GR) PX (car P) PY (cadr P))
                      (equal (list PX PY) '(4 4) 4.48)
                      (or (< (- PX (setq X (fix PX))) 0.48)
                          (< (- (setq X (fix (1+ PX))) PX) 0.48))
                      (or (< (- PY (setq Y (fix PY))) 0.48)
                          (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                      (PUNT))
                 (progn
                   (setq Q ())
                   (CLIC)
                   (if POST
                       (if V ; (CLIC) en una casella plena.
                           (if (or (member P FOTO1) (member P FOTO2))
                               (progn
                                 (command "ESPACIOP" "BORRA" "LT" ""
                                          "DESHACER" "M"
                                          "INSERT" "C" "0,0" "" "" "")
                                 (RETOL-2 "Anul-lant" "aquesta" "casella,"
                                          "caldrà" "recalcular" "solucions"
```

```
                          "compatibles" "i tornar a" "esperar...")
                    (if FOTO2
                        (progn
                          (prompt "\n.\n.")
                          (SEGUIR T)
                          (command "DESHACER" "R" "DESHACER" "M"
                                   "INSERT" "C" "0,0" "" "" "")
                          (RETOL-2 "...i, si surt" "\"EL SUDOKU"
                                 "JA TÉ UNA" "SOLUCIÓ" "ÚNICA\", mai"
                                   "no sabreu" "si sobren" "caselles"
                                   "plenes.")))
                    (initget "Si No")
                    (setq CONF (getkword (strcat "\n.\n.\nConfirmeu "
                                                 "l'anul·lació "
                                                 "(S/N)? <N>: "))
                          CONF (if CONF CONF "No"))
                    (if (= CONF "Si")
                        (setq POST ())
                        (progn
                          (command "DESHACER" "R" "UY" "ESPACIOM")
                          (CLIC))))
                  (progn
                    (setq I (assoc P S-V*V)
                          J (reverse(cdr (member I (reverse S-V*V))))
                          S-V*V (append J (cdr (member I S-V*V)))
                          I (caar S-V*V)
                          J (nth (car I) (nth (cadr I) **9*9**)))
                    (NUMCOMPAT I J ())))
              (NUMCOMPAT P N T))) ; (CLIC) en una casella buida.
        (if (not POST)
            (progn
              (if (and V (or (member P FOTO1) (member P FOTO2)))
                  (setq POST T)
                  (if (not V) (SUD-MIN **V*V** ())))
              (if POST
                  (progn
                    (if V
                        (command "DESHACER" "R")
                        (command "ESPACIOP" "BORRA" "LT" ""))
                    (command "DESHACER" "M"
                             "INSERT" "C" "0,0" "" "" "")
                    (if (or (not (or FOTO1 FOTO2)) ; En "PRE-FOTO" POST
                            (member P FOTO1)) ; activat o en "FOTO 1-2"
                        (progn  ; casella de FOTO1 afectada pel (CLIC).
                          (RETOL-2 "Determinar" "quantes"  "solucions"
                                   "normalitza-" "des són" "compatibles"
                                    "pot trigar" "hores." "Espereu...")
                          (PERFIL-V*V)
                          (COMPAT-PERFILS))
                        (progn  ; Casella de FOTO2 afectada pel (CLIC).
                          (RETOL-2 "Determinar" "quantes" "solucions"
                                   "compatibles" "hi ha" "pot trigar"
                                   "una bona" "estona." "Espereu...")
                          (ANORMALS)))
                    (command "DESHACER" "R" "UY" "ESPACIOM")))))
        (if POST
            (progn
              (setq GR (strcat "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA."
                               "\nOmplint més caselles el fareu més "
                               "fàcil (<Intro> o < > per acabar)"))
              (if (> KK 1)      ; Pot ser en "FOTO 1-2" o en "FOTO 2-2".
                  (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                                  (if KKK "" "normalitzades")
                                  " heu d'omplir més caselles:\n"))
                  (if FOTO2
                      (progn ; En "FOTO 2-2" sols 1 solució compatible.
                        (if KKK (setq FOTO2 (FOTO-V*V) KKK ()))
                        (prompt (strcat "\n.\n" GR)))
```

```
(progn ; Pasa de "FOTO 1-2" a FOTO 2-2".
       ; En "FOTO 1-2", 1 solució normal. compat.
  (prompt "\n.\nTeniu 1 solució normalitzada ")
  (prompt "compatible amb V. E. NORMALITZADA.\n")
  (SEGUIR ())
  (command "ESPACIOP" "DESHACER" "M"
           "INSERT" "C" "0,0" "" "" "")
  (RETOL-2 "Però pot" "haver-n'hi" "més, de no"
           "normalitza-" "des però" "igualment"
           "compatibles." "Així que"
           "espereu...")
  (ANORMALS)
  (command "DESHACER" "R" "UY" "ESPACIOM")
  (if (> KK 1)
      (prompt (strcat "\n.\nHi ha "
                      (itoa (1- KK))
                      " solucions més, no norma"
                      "litzades però també compa"
                      "tibles,\naixí que heu d'"
                      "omplir més caselles:\n"))
      (prompt (strcat "\n.\nCom no n'hi ha cap "
                      "de no normalitzada, "
                      GR))))))))))))
```

En alguns llocs estratègics s'han inclòs breus comentaris (que no van en negreta) per facilitar la comprensió del codi i la seva relació amb la descripció que abans s'havia fet de les etapes "PRE-FOTO", "FOTO 1-2" i "FOTO 2-2" en versió ampliada, però convindria aclarir la pluriocupació de les variables **V** i **KKK**. De la primera ja havíem dit que, a la seva definició original en **CLIC** (**(atom NV)**), n'hi havíem endossat un altre per exercir fora d'hores: la de senyal per indicar que buidàvem alguna casella de les situacions de referència (**FOTO1** o **FOTO2**) per al recompte de solucions normalitzades, de solucions ordinàries o per suspendre el recompte, i que calia actualitzar la referència, senyal que coordinàvem amb **POST** per tal que les iteracions de l'etapa "PRE-FOTO" interrompudes per **(SUD-MIN)** i les procedents de revisions fetes des de "FOTO 1-2" o "FOTO 2-2" passessin pel mateix adreçador. Sobre això, només cal afegir ara que l'adreçador es desdobla en dos tractaments específics: el tàndem **(PERFIL-V*V)** + **(COMPAT-PERFILS)** per als dos primers casos i **(ANORMALS)** per al tercer. I, pel que fa a la variable **KKK**, la missió extra que hem anunciat abans que li havíem reservat consisteix a determinar en quin moment (just quan s'assoleix el valor **KK** = **1** per primer cop després d'executar **(ANORMALS)**) cal treure la foto **FOTO2**, com podeu veure en les línies 25-28 (començant per darrera) del codi precedent, que reproduïm per a comoditat del lector:

```
(if FOTO2
    (progn ; En "FOTO 2-2" sols 1 solució compatible.
      (if KKK (setq FOTO2 (FOTO-V*V) KKK ()))
      (prompt GR))
```

Altrament, cada nova casella emplenada opcionalment donaria lloc a una altra **FOTO2** i no seria possible buidar-ne cap.

No seria correcte acabar aquest capítol donant per tancada una qüestió que no ho està gens, per bé que l'autor hagués fet abans una afirmació tan agosarada com comparar-se a Cristòfor Colom. Concretament, en la pàgina 390 es vantava d'haver descobert que la feina feta (l'estratègia de les solucions normalitzades) no només era aprofitable sinó útil i deia que bregar amb un conjunt força més reduït havia estat una bona idea, postulant implícitament que amb això s'agilitzaria el procés. Però quan es fan afirmacions així se n'ha de precisar el sentit i, si no es pot articular cap demostració concloent, almenys convé aportar indicis fragmentaris.

De primer, en la línia d'aclarir què volem significar amb això del conjunt força més reduït, cal desmuntar una presumpció que no té cap fonament però que de ben segur que va influir en l'autor a l'hora de decidir-lo a embarcar-se en l'aventura dels sudokus normalitzats. El fet que, en el conjunt de tots els sudokus possibles (parlem de SUDOKUs-SOLUCIÓ), puguem considerar subconjunts disjunts de $\mathbf{6^4 = 1.296}$ elements (el nombre de possibilitats combinant la permutació F1 de files en cada tríada amb la permutació F3 de les tríades de files) i que en cadascun d'aquests subconjunts només n'hi hagi un de normalitzat, té poca cosa a veure amb el nostre problema: a primera vista podria semblar que les solucions compatibles amb una de normalitzada també han de ser normalitzades i que, en conseqüència, n'hi ha menys

que de compatibles amb la solució sense normalitzar, però per poc que hi pensem seriosament ens adonarem que tot plegat és una bajanada i cap dels dos enunciats no té sentit. Hauríem de parlar amb propietat i, en totes dues afirmacions, fer-ho de les solucions compatibles amb la part pública d'una solució determinada, perquè parlant de solucions completes el qualificatiu "compatibilitat" no pot significar cap altre cosa que "identitat". Fent-ho així ens adonaríem que, no pas pel fet de permutar les files de cada requadre 9×3 ni de permutar aquests requadres mantenint a cada fila la posició de les caselles públiques, ha hagut de disminuir el nombre de solucions compatibles. I que les solucions compatibles amb aquestes caselles de la V. E. NORMALITZADA no tenen per què ser normalitzades, fora del cas en què, per ser pública tota la primera columna, totes les solucions compatibles la incloguin. Com tampoc no és cert que, normalitzant solucions compatibles amb la part pública de l'adoptada finalment (la VARIANT ESCOLLIDA), obtinguem miraculosament solucions compatibles amb la part pública de la seva normalitzada (la V. E. NORMALITZADA): tret dels casos en què tota la primera columna sigui pública (on sí que totes les transformacions normalitzadors respondran al mateix patró), amb caràcter general només ho podrem predicar d'aquelles en què el procediment de normalització seguit (entre els 1.296 possibles) coincideixi amb el seguit a la solució de referència; aplicant aquesta transformació concreta a les solucions esmentades en primer lloc, obtindríem totes les solucions compatibles amb la part pública de la normalitzada de referència, de les quals unes serien normalitzades (les que hem dit i potser alguna més) però la resta no. En resum: ni és cert que les solucions compatibles amb la part pública de la V. E. NORMALITZADA siguin totes normalitzades, ni que les normalitzades de les solucions compatibles amb la part pública de la VARIANT ESCOLLIDA siguin totes compatibles amb la part pública de la V. E. NORMALITZADA.

Així que, si en algun moment ens havia passat pel cap la possibilitat de fer una reducció dràstica passant dels **6.670.903.752.021.072.936.960** sudokus pronosticats per Felgenhauer i Jarvis a "només" **5.147.302.277.794.037.760** subconjunts de **1.296** sudokus, cadascun amb la seva solució normalitzada, en pressuposar que, si entre les solucions compatibles amb la part pública de la VARIANT ESCOLLIDA n'hi havia unes quantes de corresponents a un mateix subconjunt, la normalitzada d'aquest subconjunt seria compatible amb la part pública de la V. E. NORMALITZADA, i així pel que fa a la resta de solucions del primer grup, ens en podem ben bé oblidar. L'única cosa certa de tot plegat és que 1.296 VARIANTS ESCOLLIDES (que poden haver sortit d'altres tantes solucions primitives, si les transformacions aplicades eren exclusivament de tipus F1 i F3) tenen una mateixa V. E. NORMALITZADA, però això no ens serveix de res.

Desestimades aquestes presumpcions falses i proposant ara una estratègia en què de moment només considerarem solucions normalitzades però després passarem a les restants, ¿en què pot resultar útil limitar-se a les normalitzades, ni que sigui provisionalment? ¿on rau l'encert de treballar amb aquest conjunt més reduït?: doncs en les simplificacions que introduirà en el procés de cerca la manipulació exclusiva de conjunts normalitzats de files (requadres 9×3 normalitzats i tercets normalitzats d'aquests requadres). Treballar amb només una part de les solucions comporta reduir sensiblement el nombre de comparacions (saber, per exemple, que el valor de la casella **1,1** ha de ser **1**, el de la **1,4** ha de ser inferior a **5** i el de la **1,9** superior a **4**, o bé que els de **1,1**, **1,4** i **1,7** han d'estar ordenats en sentit ascendent, igual que els de **1,n**, **1,n+1** i **1,n+2**, per a **n = 1,4,7**, evita que ho haguem de comparar tot amb tot) i això, tant si preteníem tenir tots els elements inventariats com si ens conformem a reconstruir-los cada vegada, alleugereix la cerca considerablement: no és una afirmació gratuïta, sinó tan tautològica com reconèixer que, si volem destriar les boles blanques de les boles negres d'un pot, acabarem abans fent-ho només amb un grapat que estenent la selecció a tot el pot. La qüestió és si aquesta aplicació del principi *divide et vinces* amb una cerca de dues fases, la primera iterativa (**COMPAT-PERFILS**) i restringida a les solucions normalitzades compatibles, i la segona recursiva (**ANORMALS**) i aplicada a la resta de solucions compatibles (en general, unes quantes menys, perquè entre les dues fases hi haurà hagut ocasió de fer públiques més caselles i reduir així el conjunt de solucions compatibles), surt o no a compte. És a dir, si fent-ho així acabarem abans que ignorant l'artificiosa classificació entre solucions normalitzades i no normalitzades, i anant directament a la cerca de totes les solucions compatibles: **1)** exclusivament amb el procediment iteratiu **COMPAT-PERFILS**; **2)** exclusivament amb el procediment recursiu **ANORMALS**.

És això el que farem ara, però com que la disponibilitat de temps de l'autor no és ilimitada, les comparacions se centraran en un cas concret: el sudoku "difícil" de LA VANGUARDIA del 25-02-08, en les mateixes circumstàncies que ens havíen servit per introduir el tema de les solucions compatibles que queden després d'haver-nos quedat amb una sola de normalitzada, en començar aquest capítol: anirem emplenant les caselles d'aquest SUDOKU-PROBLEMA, deixant sense omplir les columnes 4ª i 5ª; quan toqui omplir la casella 4,7 amb el valor **3** (serà la 19ª) **SUD-MIN** activarà la variable **POST** i es posarà en marxa la seqüència **(PERFIL-V*V)** + **(COMPAT-PERFILS)**, després de la qual seguirem emplenant valors de la columna 4ª, deixant la casella 4,4 (**6**) pel final; amb aquesta, haurem aconseguit quedar-nos amb una única solució normalitzada compatible i es posarà en marxa **(ANORMALS)**, després de la qual només restarà omplir la casella 5,5 (**4**). És ben cert que, si haguéssim anat emplenant per files, deixant per al final la 4ª i 5ª, hauríem aconseguit retardar el moment **POST** a l'activació de la 21ª casella i els temps de processat s'haurien escurçat, però d'una banda volíem remetre el lector al supòsit inicial del capítol i, de l'altra, ja ens anava bé que els temps fossin una mica més llargs per comparar-los millor: com és lògic, els temps consignats no han estat presos cronòmetre en mà, sinó consultant en el directori actual els moments (data i hora:minuts) de creació d'uns arxius *ad hoc* que s'havien deixat programats en llocs estratègics del codi (amb dispositius com **(close (open (strcat ... ".TXT") "w")))**), sense necessitat de seleccionar-los un per un en el llistat i accedir a la informació que subministra l'opció **Propiedades** (data i hora:minuts:segons). Perquè el lector es pugui fer una composició de lloc en relació a aquestes dades, que presentem tot seguit, no està de més que sàpiga que, en efectuar les mateixes proves en els equips instal·lats a l'aula A2S105 del Campus Nord de la UPC (més endavant, en el capítol *Assegurar-se una visualització correcta*, proposem canvis en el codi perquè la utilització del programa des d'aquests equips sigui més satisfactòria) els temps baixen a un 83%.

El guió comença amb la relació de les caselles que s'han fet públiques i se situa en el moment que, en fer CLIC sobre la casella 4,7, es dispara **PERFIL-V*V**:
SUDOKU-3460 (PERFIL-V*V i COMPAT-PERFILS treballant amb solucions normalitzades)

L'activació de caselles pot haver-se fet a la finestra dreta (VARIANT ESCOLLIDA) o esquerra (V. E. NORMALITZADA), però aquí hi figuren les coordenades de la segona.
Caselles activades: 8,1 (**4**); 3,2 (**9**); 6,2 (**7**); 7,2 (**5**); 2,3 (**7**); 6,3 (**5**); 3,4 (**1**); 6,5 (**4**); 8,5 (**6**); 1,6 (**8**); 9,6 (**3**); 2,7 (**1**); 1,8 (**6**); 6,8 (**9**); 8,8 (**1**); 9,8 (**2**); 7,9 (**4**); 8,9 (**5**), que fan un total de 18.

19ª activació: 4,7 (**3**) → el requadre 3×9 central ja no té dues columnes buides.
A l'àrea gràfica, el *taijitu* és substituït pel missatge

Determinar quantes solucions normalitzades són compatibles pot trigar hores. Espereu...

A l'àrea de text, en els temps detallats en negreta, van apareixent els missatges:

0:00 *Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 196.*
0:00 *Entre 362880 línies, de compatibles amb la 2ª de la V. E. NORM. n'hi ha 76.*
0:00 *Entre 362880 línies, de compatibles amb la 3ª de la V. E. NORM. n'hi ha 324.*
0:00 *Entre 362880 línies, de compatibles amb la 4ª de la V. E. NORM. n'hi ha 628.*
0:00 *Entre 362880 línies, de compatibles amb la 5ª de la V. E. NORM. n'hi ha 340.*
0:00 *Entre 362880 línies, de compatibles amb la 6ª de la V. E. NORM. n'hi ha 168.*
0:00 *Entre 362880 línies, de compatibles amb la 7ª de la V. E. NORM. n'hi ha 92.*
0:00 *Entre 362880 línies, de compatibles amb la 8ª de la V. E. NORM. n'hi ha 40.*
0:00 *Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 96.*

0:00 *Entre 196 línies compatibles amb la 1ª, 76 amb la 2ª i 324 amb la 3ª, hi ha 10232 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*
0:05 *Entre 628 línies compatibles amb la 4ª, 340 amb la 5ª i 168 amb la 6ª, hi ha 1920 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*
0:05 *Entre 92 línies compatibles amb la 7ª, 40 amb la 8ª i 96 amb la 9ª, hi ha 2496 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

0:09 *Entre les 10232 tríades 1-2-3, les 1920 tríades 4-5-6 i les 2496 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 1117.*

[Per seguir, polseu qualsevol tecla]

A l'àrea gràfica reapareix el *taijitu* i a l'àrea de text segueixen els missatges:
Amb 1117 solucions normalitzades heu d'omplir més caselles.

20ª activació: 4,6 (**7**)
Amb 325 solucions normalitzades heu d'omplir més caselles.

21ª activació: 4,2 (**1**)
Amb 13 solucions normalitzades heu d'omplir més caselles.

22ª activació: 4,1 (**6**)
Teniu 1 solució normalitzada compatible amb V. E. NORMALITZADA.
[Per seguir, polseu qualsevol tecla]

Mentre a l'àrea de text se segueix llegint
Teniu 1 solució normalitzada compatible amb V. E. NORMALITZADA.
a l'àrea gràfica, el *taijitu* és substituït pel missatge
Però pot haver-n'hi més, de no normalitzades però igualment compatibles.
Així que espereu...

0:27 hores més tard, a l'àrea text apareix el missatge
Hi ha 592 solucions més, no normalitzades però també compatibles,
així que heu d'omplir més caselles.
a l'àrea gràfica reapareix el *taijitu* i només caldrà activar una casella més:

23ª activació: 5,2 (**4**)
EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA.
Omplint més caselles el fareu més fàcil (<Intro> o < > per acabar)

Si volem comparar aquests resultats amb els de la **alternativa 1** que abans hem suggerit (exclusivament amb el procediment iteratiu **COMPAT-PERFILS**), no costarà massa d'improvisar una versió *ad hoc*, amb un mínim de retocs.
A **PERFIL-V*V** n'hi haurà prou a substituir

```
    (foreach E (reverse (member B (reverse (member A L1))))
      (if (wcmatch E L) (setq LINIA-V*V (cons E LINIA-V*V))))
```

per

```
    (foreach E L1
      (if (wcmatch E L) (setq LINIA-V*V (cons E LINIA-V*V))))
```

(veureu que només cal modificar la primera línia). Com que només es tracta que funcioni (si aquesta versió modificada es manifestés superior, ja la poliríem), si no voleu no cal suprimir l'expressió

```
    (cond ((= K 0) (setq A "123456789" B "198765432"))
          ((= K 1) (setq A "213456789" B "897654321"))
          ((= K 2) (setq A "312456789" B "987654321"))
          ((= K 3) (setq A "213456789" B "498765321"))
          ((= K 4) (setq A "312456789" B "897654321"))
          ((= K 5) (setq A "412356789" B "987654321"))
          ((= K 6) (setq A "312456789" B "798654321"))
          ((= K 7) (setq A "412356789" B "897654321"))
          (T       (setq A "512346789" B "987654321")))
```

A **LOCT** n'hi haurà prou a substituir les cinc primeres línies

```
(defun LOCT (I / TT LL LOC)
  (if (= I 1)
      (setq TT TT1 LL '("4" "3" "2"))
      (setq TT TT2 LL '("7" "6" "5" "4" "3")))
  (foreach L LL
```

per aquestes dues

```
(defun LOCT (TT / LOC)
  (foreach L '("9" "8" "7" "6" "5" "4" "3" "2" "1")
```

Pel que fa a **COMPAT-PERFILS**, ens en sortirem substituint l'expressió

```
    (foreach T0 LL0
      (foreach T1 LL1
        (if (and (< T0 T1) (COMPAT-2L))
            (foreach T2 LL2
              (if (and (< T1 T2) (COMPAT-3L))
                  (setq TERCETS-V*V (cons (list T0 T1 T2) TERCETS-V*V)))))))
```

per

```
    (foreach T0 LL0
      (foreach T1 LL1
        (if (COMPAT-2L)
            (foreach T2 LL2
              (if (COMPAT-3L)
                  (setq TERCETS-V*V (cons (list T0 T1 T2) TERCETS-V*V)))))))
```

on veureu que ens hem limitat a suprimir les condicions **(< T0 T1)** i **(< T1 T2)**. També caldrà substituir

```
        (setq TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT 1) LOCT2 (LOCT 2) K 10)
        (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
        (foreach T0 TT0
          (if (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0)
                    L1 (SUPR (substr (cadr T0) 1 1) TOT)
                    L1 (SUPR (substr (caddr T0) 1 1) L1)
                    PRIM1 (cdr (assoc (car L1) LOCT1)))
              (mapcar '(lambda (T1 CC1)
                         (setq L2 (cdr L1) K -1 OK T) ...
```

per

```
        (setq TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        (repeat 9 (setq K (1- K) TOT (cons (itoa K) TOT)))
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
        (foreach T0 TT0
          (if (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0)
                    L1 (SUPR (substr (car T0) 1 1) TOT)
                    L1 (SUPR (substr (cadr T0) 1 1) L1)
                    L1 (SUPR (substr (caddr T0) 1 1) L1)
                    PRIM1 (cdr (assoc (car L1) LOCT1)))
              (mapcar '(lambda (T1 CC1)
                         (setq L2 (SUPR (substr (car T1) 1 1) L1) K -1 OK T) ...
```

on veureu (hem subratllat els canvis) que, a més d'adaptar els dos accessos a **LOCT** a la modificació practicada línies enrera, com que ara no és segur que la primera fila hagi de començar amb **1**, haurem d'ampliar la llista **TOT** de 8 a 9 elements per suprimir-ne el que sigui igual al valor inicial de la fila 1ª del requadre 9×3 **T0**; més endavant, com que tampoc no és segur que el valor inicial de la 1ª fila del requadre 9×3 **T1** sigui el valor més baix dels que li resten a **TOT** un cop suprimits els valors inicials de les tres files de **T0**, tampoc no podrem permetre'ns dreceres i haurem d'esbrinar amb quin valor comença aquesta fila, igual com ho fem amb les altres dues. En definitiva, i reprenent la visió de conjunt de **COMPAT-PERFILS** pel que fa a l'estratègia de comparacions comentada dos capítols enrera:

- Als efectes de veure en quin requadre **T1** = **PRIM1** de la llista **TT1** (que, tal i com ha estat creada, té ordenats aquests requadres 9×3 per valors creixents del primer element) hem de començar a analitzar la compatibilitat amb els requadres **T0** de **TT0**, buscant en el directori **LOCT1** on comencen els que tenen com a primer element el més baix dels valors **1**, **2**, **3**... **9** de la llista **L1**, de la qual han estat exclosos prèviament els valors inicials de les files 1ª, 2ª i 3ª de **T0**, ara ja no podem donar per fet que el valor inicial de la 1ª fila de **T0** sigui **1**, limitant-nos a eliminar els valors inicials de la 2ª i 3ª fila d'una lista coixa **2**, **3**... **9**, sinó que caldrà localitzar el valor **(substr (car T0) 1 1)** i treure'l explícitament d'una **L1** abans.
- Als efectes de veure en quin requadre **T2** = **PRIM2** de la llista **TT2** (que, tal i com ha estat creada, té ordenats aquests requadres 9×3 per valors creixents del primer element) hem de començar a analitzar la compatibilitat amb els requadres **T1** de **TT1**, buscant en el directori **LOCT2** on comencen els que tenen com a primer element el més baix dels valors **1**, **2**, **3**... **9** de la llista **L2**, de la qual han estat exclosos prèviament els valors inicials de les files 1ª, 2ª i 3ª de **T0** (resultant **L1**, com s'ha dit en el punt precedent) i de **T1**, ara ja no podem donar per fet que el valor inicial de la 1ª fila de **T1** sigui el primer element de **L1**, definint d'entrada **L2** com una **L1** desprovista de l'element inicial, sinó que caldrà localitzar el valor **(substr (car T1) 1 1)** i treure'l explícitament d'una **L2** que agafa el relleu d'una **L1** sencera.

Substituïdes aquestes versions modificades de **PERFIL-V*V** i **COMPAT-PERFILS** en la penúltima de **FES-PUBLIC** (la del capítol precedent), i tal i com havíem anunciat, abans de conèixer l'eficiència d'aquest plantejament no hem volgut dedicar-nos a fer neteja de tot allò relacionat amb la normalització de solucions, no fos cas que la precipitació desemboqués una vegada més en una absoluta pèrdua de temps. Aquest cop la prudència ha estat recompensada, perquè els resultats no podien ser més decebedors pel que fa a l'anomenada **alternativa 1** i, per ara, més favorable a la nostra tesi. Veieu, si no, el guió que comença amb la relació de les caselles públiques i el moment en què un CLIC sobre la casella 4,7 dispara **PERFIL-V*V: *SUDOKU-3461 (PERFIL-V*V i COMPAT-PERFILS treballant amb directament amb solucions ordinàries) aplicat sobre **V*V** (V. E. NORMALITZADA)***
L'activació de caselles pot haver-se fet a la finestra dreta (VARIANT ESCOLLIDA) o esquerra (V. E. NORMALITZADA), però aquí hi figuren les coordenades de la segona.
Caselles activades: 8,1 (**4**); 3,2 (**9**); 6,2 (**7**); 7,2 (**5**); 2,3 (**7**); 6,3 (**5**); 3,4 (**1**); 6,5 (**4**); 8,5 (**6**); 1,6 (**8**); 9,6 (**3**); 2,7 (**1**); 1,8 (**6**); 6,8 (**9**); 8,8 (**1**); 9,8 (**2**); 7,9 **(4)**; 8,9 (**5**), que fan un total de 18.

19ª activació: 4,7 (**3**) → el requadre 3×9 central ja no té dues columnes buides. A l'àrea gràfica, el *taijitu* és substituït pel missatge

> *Determinar quantes solucions normalitzades són compatibles pot trigar hores. Espereu...*

A l'àrea de text, en els temps detallats en negreta, van apareixent els missatges:

0:00 *Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 902.*
0:00 *Entre 362880 línies, de compatibles amb la 2ª de la V. E. NORM. n'hi ha 100.*
0:00 *Entre 362880 línies, de compatibles amb la 3ª de la V. E. NORM. n'hi ha 573.*
0:00 *Entre 362880 línies, de compatibles amb la 4ª de la V. E. NORM. n'hi ha 1020.*
0:00 *Entre 362880 línies, de compatibles amb la 5ª de la V. E. NORM. n'hi ha 546.*
0:00 *Entre 362880 línies, de compatibles amb la 6ª de la V. E. NORM. n'hi ha 168.*
0:00 *Entre 362880 línies, de compatibles amb la 7ª de la V. E. NORM. n'hi ha 120.*
0:00 *Entre 362880 línies, de compatibles amb la 8ª de la V. E. NORM. n'hi ha 40.*
0:00 *Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 180.*

0:10 *Entre 902 línies compatibles amb la 4ª, 100 amb la 5ª i 573 amb la 6ª, hi ha 66560 tríades compatibles amb la V. E. NORMALITZADA.*
0:29 *Entre 1020 línies compatibles amb la 1ª, 546 amb la 2ª i 168 amb la 3ª, hi ha 17968 tríades compatibles amb la V. E. NORMALITZADA.*
0:29 *Entre 120 línies compatibles amb la 7ª, 40 amb la 8ª i 180 amb la 9ª, hi ha 7936 tríades compatibles amb la V. E. NORMALITZADA.*

Hores més tard, abans que s'hagi completat l'anàlisi de la compatibilitat entre les 66.560 tríades 1-2-3, les 17.968 tríades 4-5-6 i les 7.936 tríades 7-8-9, i quan l'ordinador presenta signes inequívocs d'anar ofegat de memòria, surt l'avís

> *Windows - Mínimo de memoria virtual demasiado bajo*
> *El sistema no tiene memoria virtual suficiente. Windows está aumentando el tamaño del archivo paginado de memoria virtual. Durante este proceso las peticiones de memoria de algunas aplicaciones podrán denegarse. Consulte la Ayuda para más información.*

Amb independència de les condicions de treball desfavorables que probablement han provocat el col·lapse de memòria (o l'han precipitat, abans del que seria raonable fins i tot en un equip similar), com podria ser una configuració poc adequada o una ocupació excessiva en disc, sembla indubtable que la seva causa és l'embalum que haurà assolit la llista **SUDOKUS-V*V** de solucions compatibles; com també ho és que intentar eludir el problema descarregant aquesta informació en un o diversos arxius (tot i que poguéssim escollir-ne la localització i el format) no seria una bona solució. En qualsevol cas, ha quedat "demostrada" (amb tot el relativisme del terme, atès que la "conclusió" que estem a punt de formular no deriva d'un discurs lògic ni tan sols d'una mostra representativa de casos, sinó que estem especulant sobre un únic exemple concret) la superioritat del mètode híbrid sobre l'assajat ara mateix, i no té gaire sentit divagar sobre possibles mesures correctores. Una altra cosa, que sí que pot fer llum sobre l'ordre de magnitud de la relació entre solucions normalitzades i solucions ordinàries en el moment d'activació de **POST**, és que vulguem saber quantes n'hi hauria de les segones, enfront de les **1.117** de normalitzades. Doncs bé: si a **COMPAT-PERFILS** modifiquem l'expressió

```
                                        (if OK (setq SUDOKUS-V*V
                                                     (cons (SUD-NUM) SUDOKUS-V*V)
                                                     S-V*V (cons T S-V*V))))
```
eliminant tota referència a **SUDOKUS-V*V** i deixant-la en
```
                                        (if OK (setq S-V*V (cons T S-V*V))))
```
i, havent-nos tret el mort de sobre, tornem a executar l'última versió probada, al cap de **16:27** hores el programa ens dirà que

> *Entre les 66560 tríades 1-2-3, les 17968 tríades 4-5-6 i les 7936 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha **232683**.*

Dit això, ens podria quedar el dubte de si els resultats variarien molt aplicant **PERFIL-V*V** i **COMPAT-PERFILS** a ****V**V**** (la part pública de la VARIANT ESCOLLIDA) en comptes de fer-ho a ****V*V**** (la part pública de la V. E. NORMALITZADA). Doncs no ens en hem de privar, perquè serà tan senzill com substituir ****V**V**** per ****V*V**** en les seves aparicions en aquestes dues funcions i també a **FOTO-V*V** (que no seria sobrer anomenar ara **FOTO-V**V**, igual que reanomenar com a **PERFIL-V**V** la funció **PERFIL-V*V** i com a **LINIA-V**V**, **LINIES-V**V**, **TERCETS-V**V**, **SUDOKUS-V**V** i **S-V**V** les variables **LINIA-V*V**, **LINIES-V*V**, **TERCETS-V*V**, **SUDOKUS-V*V** i **S-V*V**).
El guió d'execució, en les mateixes condicions del precedent, serà:
SUDOKU-3462 (PERFIL-V**V i COMPAT-PERFILS treballant amb directament amb solucions ordinàries) aplicat sobre **V**V** (VARIANT ESCOLLIDA)
Com que en aquesta versió sols juguem amb la VARIANT ESCOLLIDA, sense normalitzar, les coordenades responen a aquesta opció.
Caselles activades: 1,1 (**8**); 9,1 (**3**); 3,2 (**1**); 6,3 (**4**); 8,3 (**6**); 8,4 (**4**); 3,5 (**9**); 6,5 (**7**); 7,5 (**5**); 2,6 (**7**); 6,6 (**5**); 2,7 (**1**); 7,8 (**4**); 8,8 (**5**); 1,9 (**6**); 6,9 (**9**); 8,9 (**1**); 9,9 (**2**), que fan un total de 18.

19ª activació: 4,7 (**3**) → el requadre 3×9 central ja no té dues columnes buides.
A l'àrea gràfica, el *taijitu* és substituït pel missatge

> *Determinar quantes solucions normalitzades són compatibles pot trigar hores. Espereu...*

A l'àrea de text, en els temps detallats en negreta, van apareixent els missatges:

0:00 *Entre 362880 línies, de compatibles amb la 1ª de la VARIANT E. n'hi ha 168.*
0:00 *Entre 362880 línies, de compatibles amb la 2ª de la VARIANT E. n'hi ha 1020.*
0:00 *Entre 362880 línies, de compatibles amb la 3ª de la VARIANT E. n'hi ha 546.*
0:00 *Entre 362880 línies, de compatibles amb la 4ª de la VARIANT E. n'hi ha 902.*
0:00 *Entre 362880 línies, de compatibles amb la 5ª de la VARIANT E. n'hi ha 100.*
0:00 *Entre 362880 línies, de compatibles amb la 6ª de la VARIANT E. n'hi ha 573.*
0:00 *Entre 362880 línies, de compatibles amb la 7ª de la VARIANT E. n'hi ha 120.*
0:00 *Entre 362880 línies, de compatibles amb la 8ª de la VARIANT E. n'hi ha 180.*
0:00 *Entre 362880 línies, de compatibles amb la 9ª de la VARIANT E. n'hi ha 40.*

0:10 *Entre 168 línies compatibles amb la 4ª, 1020 amb la 5ª i 546 amb la 6ª, hi ha 17968 tríades compatibles amb la VARIANT ESCOLLIDA.*
0:21 *Entre 902 línies compatibles amb la 1ª, 100 amb la 2ª i 573 amb la 3ª, hi ha 66560 tríades compatibles amb la VARIANT ESCOLLIDA.*
0:22 *Entre 120 línies compatibles amb la 7ª, 180 amb la 8ª i 40 amb la 9ª, hi ha 7936 tríades compatibles amb la VARIANT ESCOLLIDA.*

Hores més tard, abans que s'hagi completat l'anàlisi de la compatibilitat entre les 17.968 tríades 1-2-3, les 66.560 tríades 4-5-6 i les 7.936 tríades 7-8-9, i quan l'ordinador presenta signes inequívocs d'anar ofegat de memòria, de nou surt l'avís de la versió precedent.

Veureu que, pel que fa a l'avaluació de línies i requadres 9×3 compatibles, abans de l'avaluació de solucions ordinàries compatibles que desemboca en col·lapse, el resultat millora sensiblement en aquest segon cas. Però això no deixa de ser una curiositat irrellevant, el sentit de la qual probablement s'invertiria (resultat millor processant ****V*V**** en comptes de ****V**V****) en un cas particular diferent. I, si volem eliminar **SUDOKUS-V**V** de la definició de COMPAT-PERFILS, com ho havíem fet amb **SUDOKUS-V*V** a la versió precedent, no cal molestar-se a repetir l'assaig, perquè el missatge final previsiblement tornarà a ser

> *Entre les 17968 tríades 1-2-3, les 66560 tríades 4-5-6 i les 7936 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha **232683**.*

Ara potser és el moment de fer una pausa i, conscients la gairebé nul·la capacitat de generalització atribuïble a l'estudi d'un únic cas particular, aproximar-nos a una qüestió que havia quedat sense resposta: la relació que hi pugui haver entre el total de solucions compatibles amb un conjunt caselles públiques i el nombre de solucions normalitzades compatibles amb aquest conjunt. Per a més inri, la qüestió no queda contestada amb un únic valor numèric per a cada cas concret (cada conjunt de caselles públiques al límit de l'activació de **POST**) ja que, per tenir-lo ben apamat, caldria formar un conjunt d'arbres (un per a cada casella no emplenada) en què cada node constaria de 6 enters: la referència al node precedent; el parell de coordenades que identifica alguna de les caselles encara no emplenades; un valor de 1 a 9, entre els que admet la casella en última instància (en el sentit donat a aquesta expressió parlant de **RE-MEMBER** < **MASCARA** < **OMPLE-SUDOKU** o de **RE+MEMBER** < < **ANORMALS**); el nombre total de solucions compatibles amb les caselles emplenades i el nombre de solucions normalitzades compatibles, fins a tenir-ne només una. Tanmateix, aquí ens limitem a una branca d'un d'aquests arbres i, a sobre, només aportem les dades completes de l'arrel i de la fulla, perquè dels nodes intermedis en desconeixem el nombre de solucions compatibles. Però, com que l'únic que pretén l'autor és extreure alguna conclusió dels assajos realitzats per tal que el lector entusiasta pugui ampliar-los fins a on la seva paciència li permeti i, si més no, per tal de situar-lo en l'ordre de magnitud de les xifres en què ens movem, vet aquí aquestes dades ben endreçades:

	casella	M solucions compatibles	N s. c. normalitzades	ràtio M/N
- node 1:	4,7 (**3**)	232.687	1.117	208,31
- node 2:	4,6 (**7**)	- - - -	325	- - -
- node 3:	4,2 (**1**)	- - - -	13	- - -
- node 4:	4,1 (**6**)	593	1	593,-

No hi ha cap raó per afirmar que, amb les dades completes, els valors de l'última columna estarien ordenats en sentit creixent, però sí que és plausible suposar que no saltarien a un ordre de magnitud molt diferent. Si el lector ho vol verificar, només caldrà que practiqui les modificacions suggerides darrerament per no limitar a les normalitzades la detecció de solucions compatibles i per evitar el col·lapse de memòria, i que hi afegeixi petits canvis a la definició de la funció **SUD-MIN**: a la línia de codi

```
(if (and J1 POST (< I 17)) (setq POST ())
```

caldrà que la primera vegada substitueixi el valor **17** per **20**, per poder activar la casella 4,6 (**7**) abans que s'activi el dispositiu **(PERFIL-V*V)** + **(COMPAT-PERFILS)**, i que la segona el substitueixi per **21**, per poder activar la casella 4,2 (**1**), en la seguretat que... la resposta arribarà abans de les 16:27 hores d'espera que va haver de suportar l'autor en l'últim assaig descrit!

Havent vist què passa amb l'**alternativa 1**, no es gens difícil de pronosticar què passarà amb l'**alternativa 2** (recordem-ho: jugar exclusivament amb el procediment recursiu **ANORMALS**, així que sigui **POST** = **T**), ara que sabem que cal rastrejar fins haver detectat 232.687 solucions compatibles: l'única diferència serà que abans d'un col·lapse llargament anunciat (salvant la llicència idiomàtica i la paradoxa) l'anunci serà molt més parc que abans, perquè després d'aparèixer a l'àrea gràfica l'avís *Determinar quantes solucions compatibles hi ha pot trigar hores. Espereu...* ja no n'hi haurà cap més i directament es farà el silenci fins que, hores després (i ara ja tant se val si en són més o menys), es consumi la predicció. Si ho voleu comprovar, només caldrà que a **FES-PUBLIC** substituïu

```
                                (progn  ; casella de FOTO1 afectada pel (CLIC).
                                  (RETOL-2 "Determinar" "quantes"  "solucions"
                                           "normalitza-" "des són" "compatibles"
                                            "pot trigar" "hores." "Espereu...")
                                  (PERFIL-V*V)
                                  (COMPAT-PERFILS))
```

per

```
                                (progn  ; casella de FOTO1 afectada pel (CLIC).
                                  (RETOL-2 "Determinar" "quantes"  "solucions"
                                           "compatibles" "hi ha" "pot trigar"
                                           "hores." "" "Espereu...")
                                  (ANORMALS))
```

Aquí només ens hem molestat a canviar els arguments de **RETOL-2**, per ser coherents amb l'avís esmentat però sobretot perquè no us confonguéssiu amb l'expressió **progn** que venia a continuació en el codi, amb un **(ANORMALS)** que ja mai no s'executarà.

Malgrat la superioritat indiscutible (si més no, pel que fa al cas de referència) que reporta la divisió en dues etapes del procés de cerca de solucions compatibles (detecció de solucions normalitzades, de primer, i de les solucions ordinàries que complementen la normalitzada única, després d'activar unes quantes caselles més), els 27 minuts de la segona, que en l'exemple representa les tres quartes parts del temps d'espera (**9 + 27 = 36** minuts, excloent-ne el d'activació de més caselles que dóna accés a aquesta segona etapa), resulta excessiu... sobretot tenint en compte que l'avantatge aconseguit d'una banda en arribar a només una solució normalitzada compatible, el malbaratem de l'altra: de fet, la recursió **RE+MEMBER** actua cegament en la cerca de solucions compatibles perquè no n'exclou les normalitzades, i en el mateix sentit la funció **ANORMALS** no compleix allò que el seu nom prometia. I, que ningú no es precipiti, argumentant amb suficiència que de solucions normalitzades ja només en queda una i que una fotesa així (1 entre 593, a l'exemple) gairebé no repercutirà a la durada total. Perquè una cosa són les solucions que cal examinar (totes les determinades pels valors admissibles en primera instància a cadascuna de les caselles desactivades, just abans de posar-se en marxa **RE+MEMBER**, entre les quals n'hi haurà de normalitzades i de no normalitzades) i una altra les que hagin resultat compatibles amb el conjunt de caselles plenes, en haver pogut completar la cadena recursiva (només un d'aquests tests reixits correspondrà a la solució normalitzada que ha iniciat l'exploració; la resta, a solucions no normalitzades). Com que totes les del primer grup han de passar per l'adreçador, consumint més o menys temps de processat encara que finalment no hagin d'acabar en el segon, però *a priori* ja sabem que de normalitzades només en prosperarà una, estalviaríem temps detectant les solucions normalitzades quan abans millor i retirant-les d'escena.

Doncs això mateix és el que fem en les versions millorades de **RE+MEMBER** i **ANORMALS** que mostrem tot seguit, el secret de les quals rau en una senzilla propietat que ens permetrà la detecció precoç dels sudokus normalitzats (només caldrà conèixe'n la primera fila) per posar-los fora de circulació: transposant les caselles d'un sudoku normalitzat, com si hi apliquéssim la transformació TR, la primera columna passa a ser primera fila

$$a_{11}\quad a_{21}\quad a_{31}\qquad a_{41}\quad a_{51}\quad a_{61}\qquad a_{71}\quad a_{81}\quad a_{m1}$$

i s'ha d'acomplir $a_{11} < a_{21} < a_{31}$, $a_{41} < a_{51} < a_{61}$, $a_{71} < a_{81} < a_{91}$ i $a_{11} < a_{41} < a_{71}$.

```
(defun RE+MEMBER (R-9*9 R-LLL / 2R R-N R-P F C0 C3 C6)
  (if (COMPLET-9*9 R-9*9)
      (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9) SUDOKUS-V*V)
            KKK (cons T KKK))
      (progn
        (if (and PRE (> (cadr R-P) 0))
            (setq PRE () F (nth 0 R-9*9)
                  C0 (nth 0 F) C3 (nth 3 F) C6 (nth 6 F)
                  NORM (and (< C0 C3 C6)
                            (< C0 (nth 1 F) (nth 2 F))
                            (< C3 (nth 4 F) (nth 5 F))
                            (< C6 (nth 7 F) (nth 8 F)))))
        (if (or PRE (not NORM))
            (progn
              (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
              (foreach E L1A9
                (if (member E R-N)
                    (progn
                      (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                      (RE+MEMBER (car 2R) (cadr 2R))))))
            (if NORM (setq PRE T))))))

(defun ANORMALS (NORM=1 / L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P PRE NORM)
  (setq L0A8 '(0 1 2 3 4 5 6 7 8) FOTO1 () FOTO2 (FOTO-V*V))
  (if (not NORM=1) (setq KKK () SUDOKUS-V*V ()))
  (foreach X L0A8
    (setq XX ())
    (foreach Y L0A8
      (if (atom (setq V (ELEMENT **V*V**))) (setq XX (cons V XX))))
    (setq XXX (cons XX XXX)))
  (setq XXX (reverse XXX))
```

```
  (foreach Y L0A8
    (setq YY ())
    (foreach X L0A8
      (if (atom (setq V (ELEMENT **V*V**)))
          (setq YY (cons V YY))))
    (setq YYY (cons YY YYY)))
  (setq YYY (reverse YYY))
  (foreach YY '(0 3 6)
    (foreach XX '(0 3 6)
      (setq L () Y -1)
      (foreach VV **V*V**
        (setq Y (1+ Y) X -1)
        (if (not (or (< Y YY) (> Y (+ YY 2))))
            (progn
              (foreach V VV
                (setq X (1+ X))
                (if (not (or (< X XX) (> X (+ XX 2))))
                    (if (atom V) (setq L (cons V L))))))))
      (setq XY (cons L XY))))
  (setq XY (reverse XY))
  (foreach Y L0A8
    (setq LL ())
    (foreach X L0A8
      (setq L ())
      (if (listp (ELEMENT **V*V**))
          (foreach E L1A9
            (setq L (cons (if (or (member E (nth X XXX)) (member E (nth Y YYY))
                                  (member E (nth (+ (/ X 3) (* (/ Y 3) 3)) XY)))
                              () E)
                          L))))
      (setq LL (cons (reverse L) LL)))
    (setq LL (reverse LL) LLL (cons LL LLL)))
  (setq LLL (reverse LLL))
  (if NORM=1
      (progn
        (setq PRE T LL ())
        (foreach EE (TRANSPOSAR **V*V**)
          (setq L ())
          (foreach E EE (setq L (cons (if (atom E) E (reverse E)) L)))
          (setq LL (cons (reverse L) LL)))
        (setq LL (reverse LL))
        (RE+MEMBER LL (TRANSPOSAR LLL)))
      (RE+MEMBER **V*V** LLL))
  (setq KK (length KKK) KKK (cons P KKK) S-V*V (list KKK)))
```

Observeu que ara **ANORMALS** incorpora l'argument binari **NORM=1**, perquè el dispositiu de detecció i abandonament de sudokus normalitzats emergents només cal activar-lo quan s'hi accedeix per primer cop (etapa "FOTO 1-2"). Per contra, quan hi accedim per segona vegada (etapa "FOTO 2-2") perquè la desactivació d'una casella ens fa retornar a una situació en què entre les solucions compatibles n'hi poden haver de normalitzades, no n'hem de descartar cap i **RE+MEBER** haurà de rastrejar com abans. Per a no fer-ho massa llarg ens limitarem a localitzar les dues crides a **ANORMALS**, subratllant-les en una versió de **FES-PUBLIC** que hem abreviat mitjançant l'omissió de diversos fragments de codi irrellevants a aquests efectes, que s'han substituït per punts suspensius:

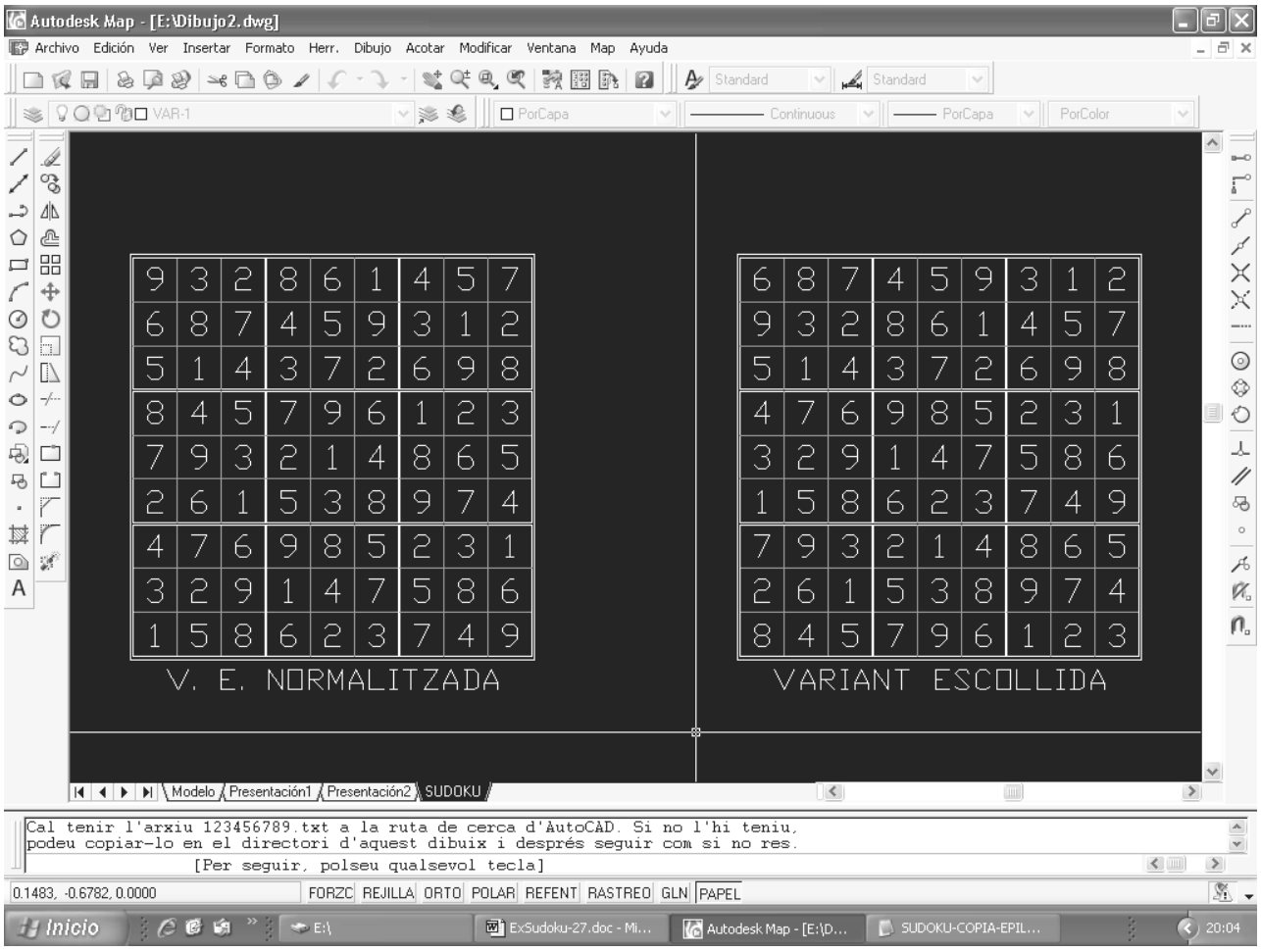
```
(defun FES-PUBLIC (/ KK KKK Q QX QY NV V GR L1 SUDOKUS-V*V S-V*V FOTO1 FOTO2 *K*
                     *KK*)
  (setq **V*V** (INI-COORDS) **V**V** **V*V**)
  (prompt "\n.\n.\n.")
  (graphscr)
  (command "ESPACIOM" "CVPORT" 2 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""
           "CVPORT" 3 "CAPA" "D" "NORM-1" ""
           "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  (YIN-YANG)
```

```
(while (not (or ...))
       (if (= (car GR) 5)
           (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
           (if (and ...)
               (progn
                 (setq Q ())
                 (CLIC)
                 (if POST ...)
                 (if (not POST)
                     (progn
                       (if (and V (or (member P FOTO1) (member P FOTO2)))
                           (setq POST T)
                           (if (not V) (SUD-MIN **V*V** ())))
                       (if POST
                           (progn
                             (if V
                                 (command "DESHACER" "R")
                                 (command "ESPACIOP" "BORRA" "LT" ""))
                             (command "DESHACER" "M"
                                      "INSERT" "C" "0,0" "" "" "")
                             (if (or (not (or FOTO1 FOTO2)) ; En "PRE-FOTO" POST
                                     (member P FOTO1)) ; activat o en "FOTO 1-2"
                                 (progn ...)
                                 (progn  ; Casella de FOTO2 afectada pel (CLIC).
                                   (RETOL-2 "Determinar" "quantes" "solucions"
                                            "compatibles" "hi ha" "pot trigar"
                                            "una bona" "estona." "Espereu...")
                                   (ANORMALS ())))
                             (command "DESHACER" "R" "UY" "ESPACIOM")))))
                 (if POST
                     (progn
                       (setq GR (strcat "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA."
                                        "\nOmplint més caselles el fareu més "
                                        "fàcil (<Intro> o < > per acabar)"))
                       (if (> KK 1)     ; Pot ser en "FOTO 1-2" o en "FOTO 2-2".
                           (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                                           (if KKK "" "normalitzades")
                                           " heu d'omplir més caselles:\n"))
                           (if FOTO2
                               (progn ...)
                               (progn ; Pasa de "FOTO 1-2" a FOTO 2-2".
                                      ; En "FOTO 1-2", 1 solució normal. compat.
                                 (prompt "\n.\nTeniu 1 solució normalitzada ")
                                 (prompt "compatible amb V. E. NORMALITZADA.\n")
                                 (SEGUIR ())
                                 (command "ESPACIOP" "DESHACER" "M"
                                          "INSERT" "C" "0,0" "" "" "")
                                 (RETOL-2 "Però pot" "haver-n'hi" "més, de no"
                                          "normalitza-" "des però" "igualment"
                                          "compatibles." "Així que" "espereu...")
                                 (ANORMALS T)
                                 (command "DESHACER" "R" "UY" "ESPACIOM")
                                 (if (> KK 1) ...)))))))))))
```

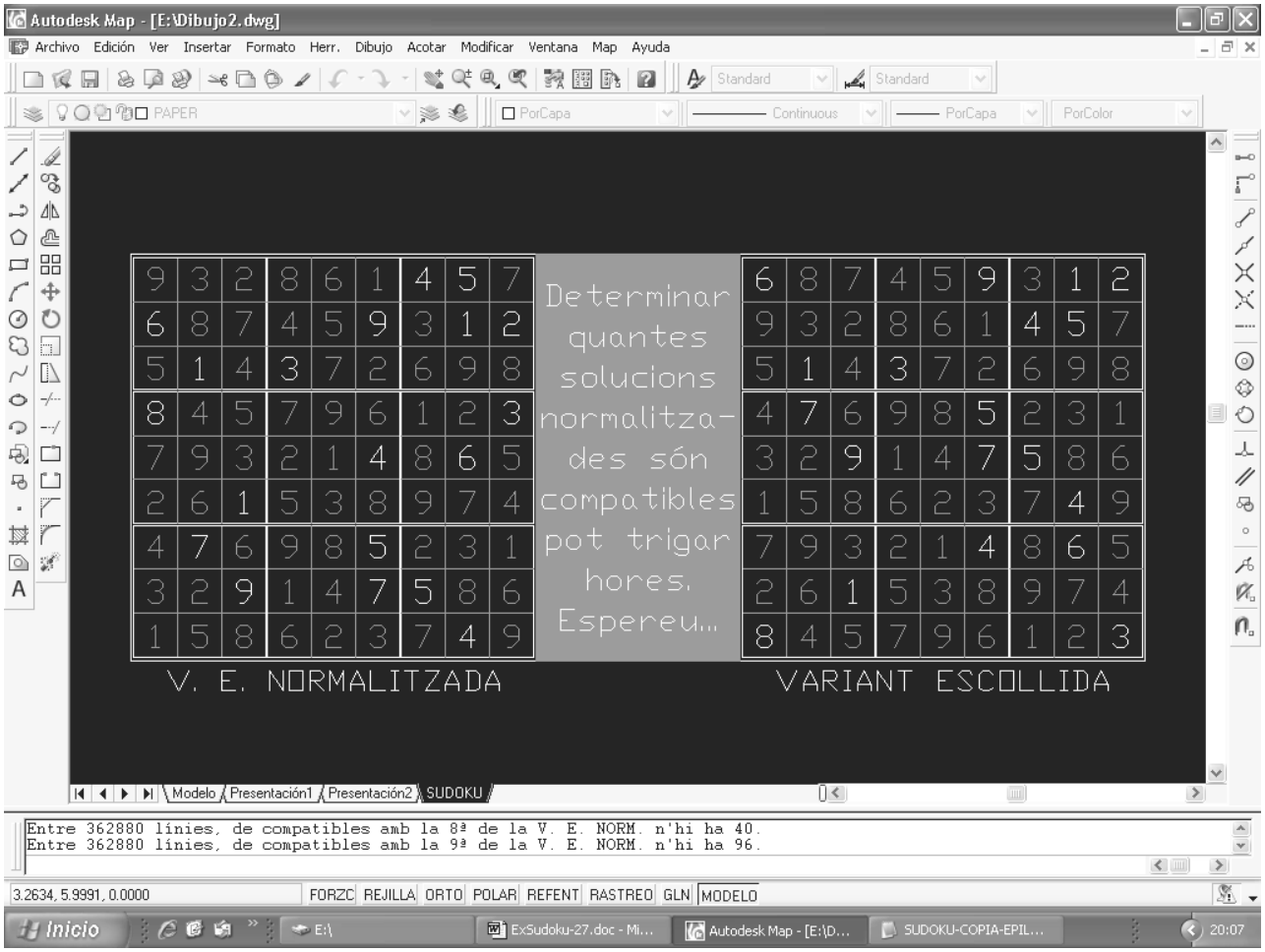
El refinament ha donat els seus fruits, perquè hem aconseguit fer baixar a menys de la meitat el temps de cerca de les solucions ordinàries que complementen la normalitzada única: en concret, de **27** a **11** minuts. És a dir que ara aquesta segona etapa representa poc més de la meitat de la durada total del procés, que ha baixat de **9 + 27 = 36** a **9 + 11 = 20** minuts (o més exactament, a **8'50" + 11'40" = 20'30"**).

****
****
****

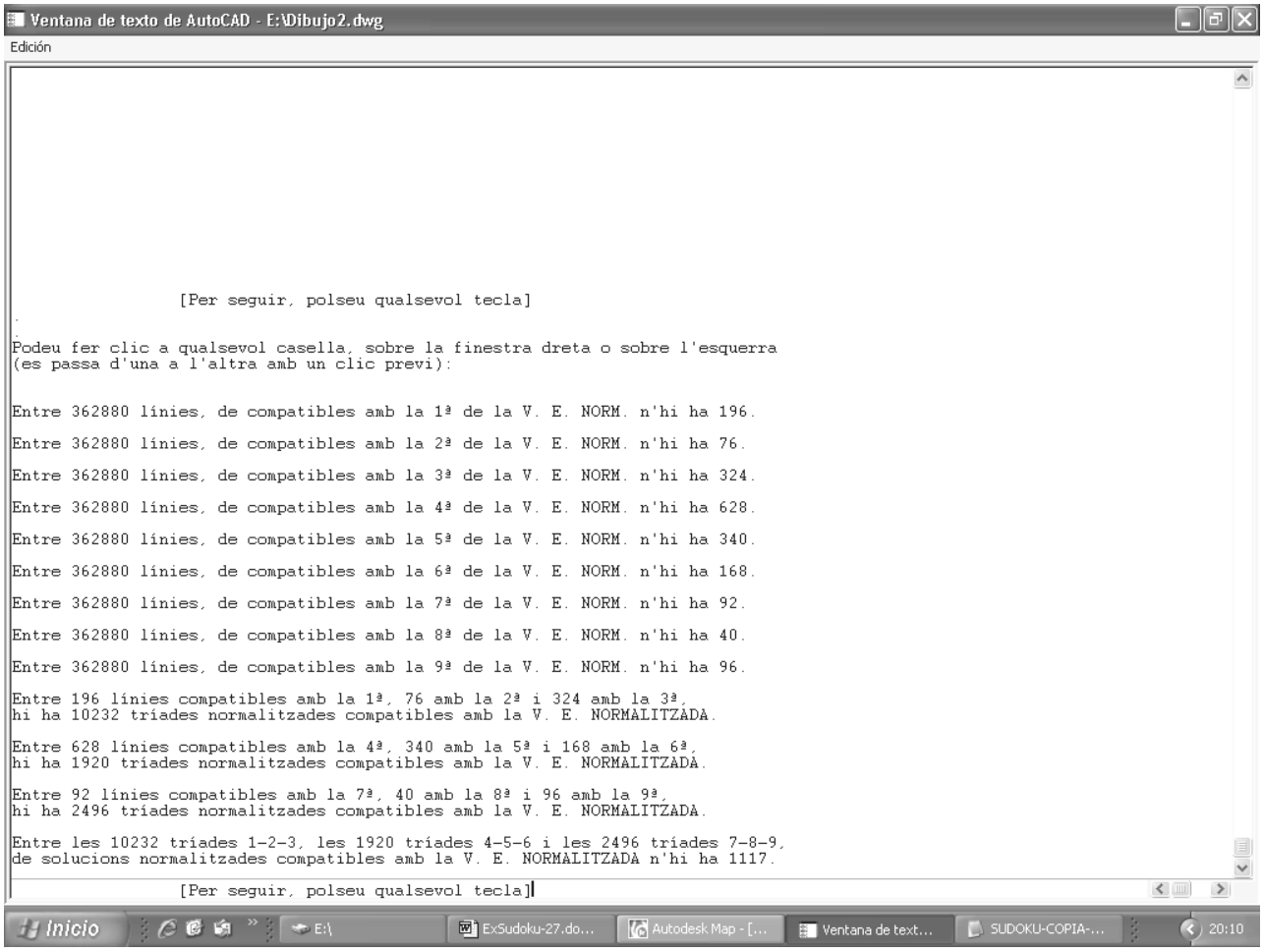
En el bloc d'actualitzacions que clou el capítol següent (concretament, entre les pàgines 473 i 483) veureu com aconseguim passar del temps 8'50" + 11'40" = 20'30" esmentat a 3'10" + 0'20" = 3'30".



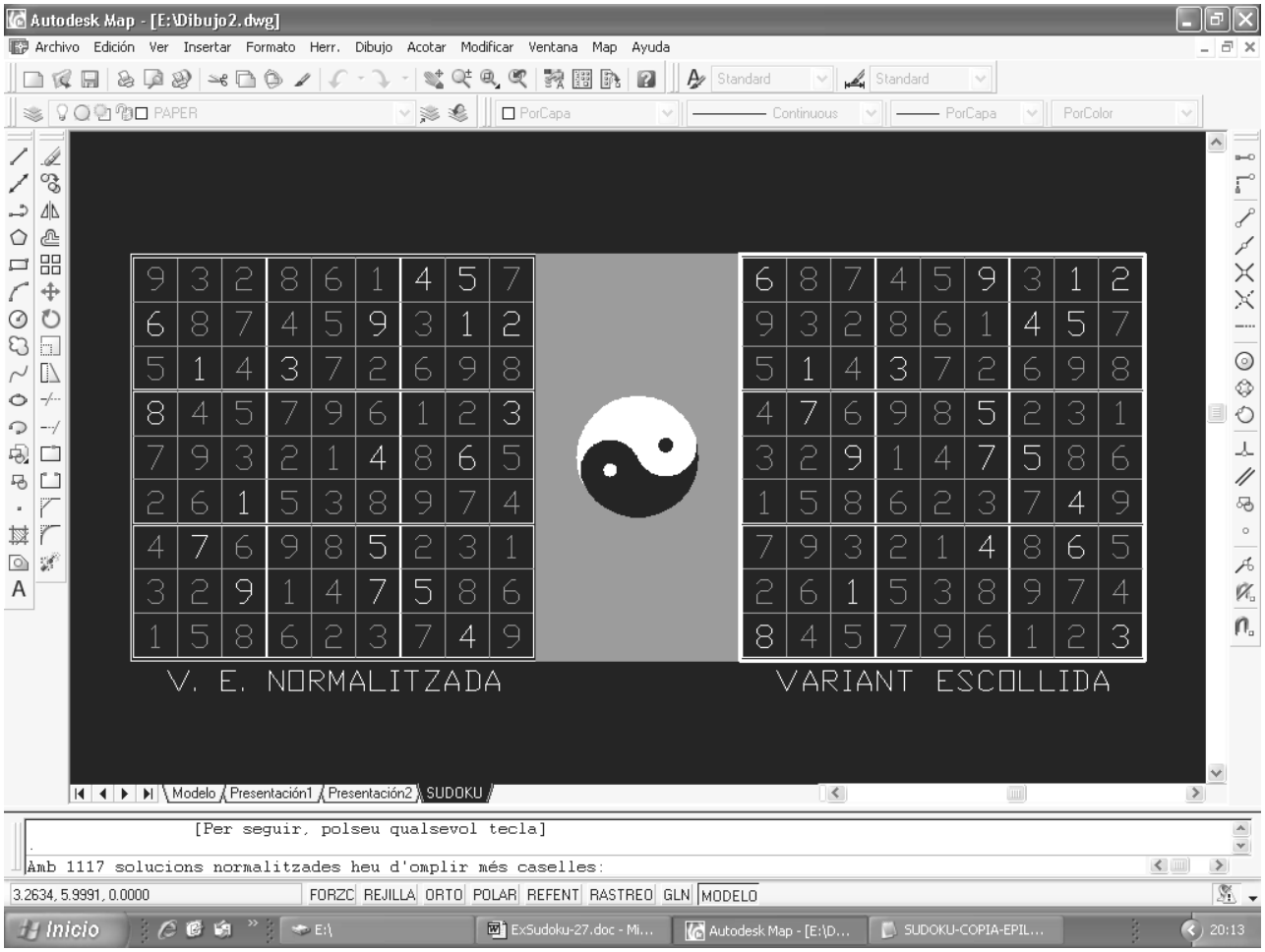
Un cop normalitzada la solució del sudoku difícil de LA VANGUARDIA (p. 145),...



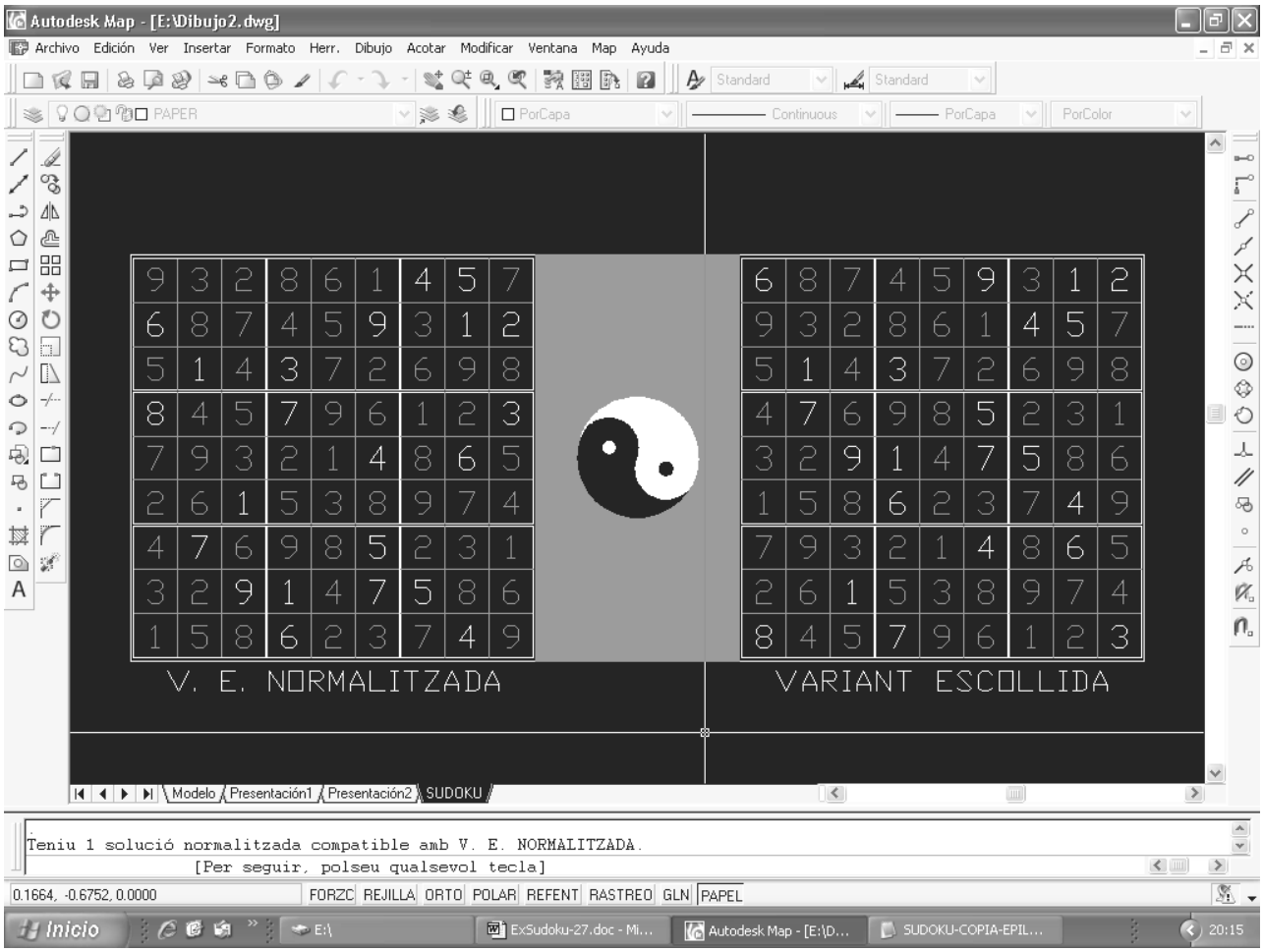
activarem caselles (les farem públiques). Amb aquestes 19, es posarà en marxa...



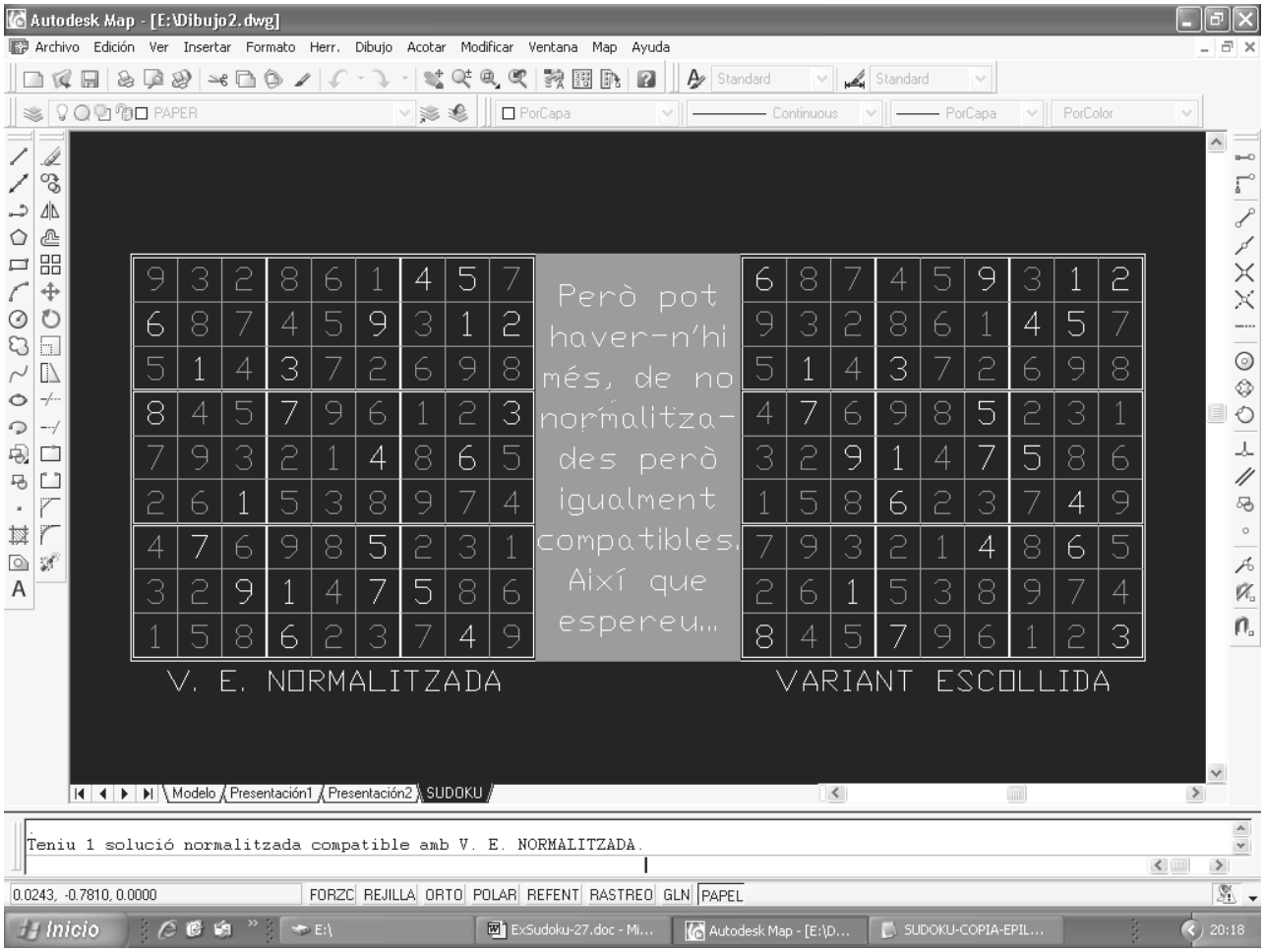
el còmput de solucions normalitzades compatibles, que ens ofereix aquest resultat.



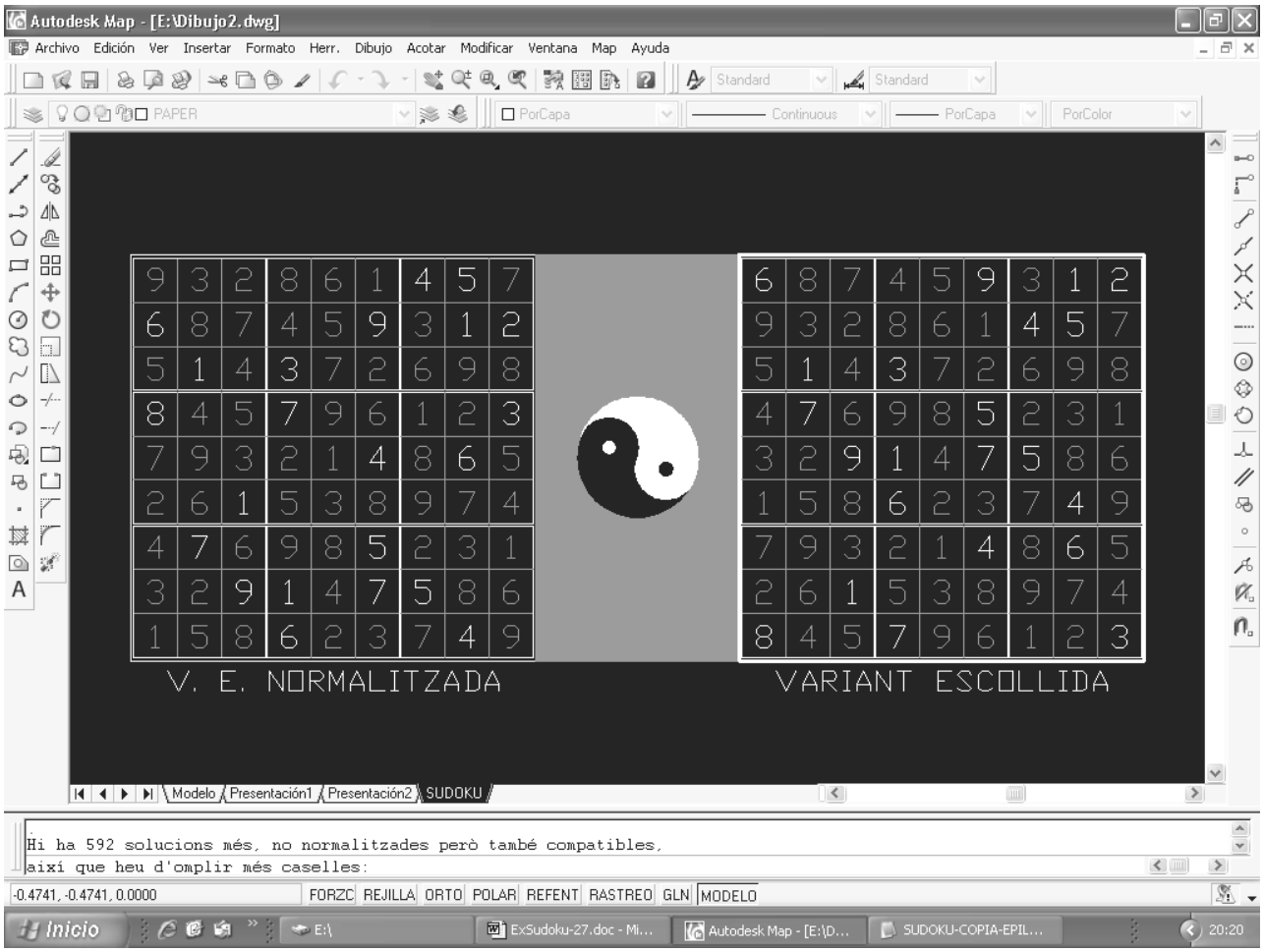
Segons SUDÒKULUM, cal seguir activant caselles. En farem públiques algunes més...



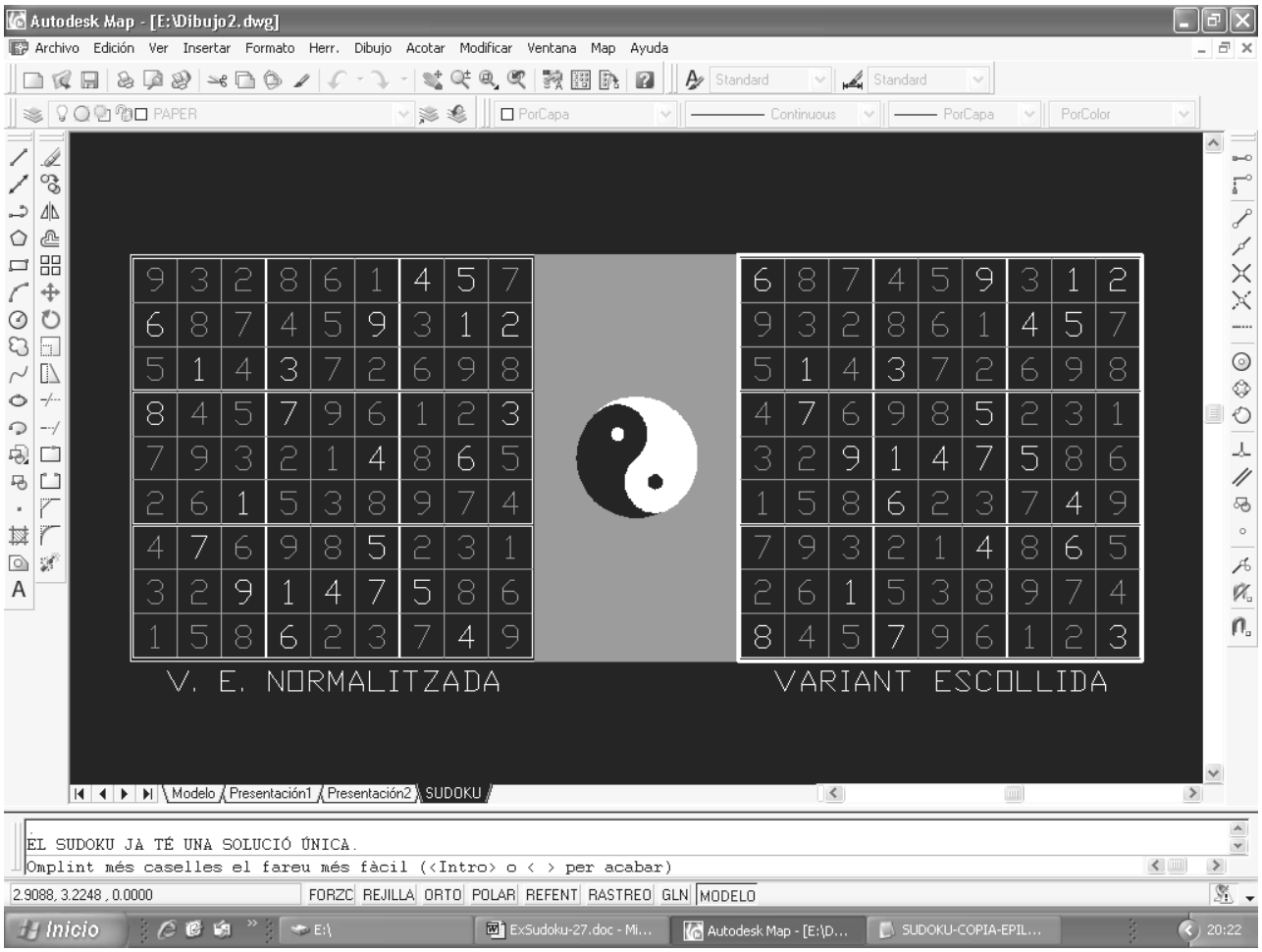
fins que, quan portem activades aquestes 22, diu que només queda una solució...



normalitzada compatible (la que veiem a l'esquerra, en gris) però que n'hi ha...



més, de compatibles, i que hem d'activar més caselles. Gràcies a LA VANGUARDIA...



sabem que activant la 5,5 s'haurà acabat (també hauríem acabat activant la 1,6).

Un nou camí i també una drecera

De bon principi havia quedat clar que aquest era un programa per crear sudokus, no pas per resoldre'ls. Però a ningú que hagi llegit *Etapa prèvia: crear una solució* li haurà passat per alt que l'instrument també podria servir per a una missió tan absurda com la segona (absurda en el sentit d'autoaxafar-nos la guitarra, privant-nos de l'entreteniment de resoldre'ls manualment): en lloc d'utilitzar l'opció A per improvisar una SUDOKU-SOLUCIÓ i extreure'n un SUDOKU-PROBLEMA o més, podríem executar **OMPLE-SUDOKU** introduint-hi els valors declarats d'un SUDOKU-PROBLEMA i, quan els haguéssim acabat de passar, seguir omplint caselles, usant en cadascuna l'únic valor amb què se'ns permetés de fer-ho (si el SUDOKU-PROBLEMA era correcte només hi hauria una solució i, en conseqüència, **MASCARA** només ens mostraria un nombre per a cada assignació). Només hauríem de substituir **SUD-MIN** per una versió *ad hoc* perquè **POST** no s'activés fins haver transferit tots els valors del sudoku: una cosa tan senzilla com reduir els criteris que intervenen en la determinació del llindar al nombre de caselles plenes, que faríem coincidir amb el del sudoku; a efectes pràctics n'hi hauria prou a substituir el valor **17** pel cardinal d'aquest sudoku, a la 4ª línia de **SUD-MIN** (i a realitzar la operació inversa per restituir al programa la funcionalitat original, una vegada enllestida l'aplicació atípica). També dins d'**ACTUALITZA**, i tant amb el propòsit d'agilitzar el processat com de retardar l'activació de **POST**, impedint que pogués produir-se abans que el **SUD-MIN** de conveniència ho provoqués, podríem prescindir de la funció de detecció **T+D<12**. Tot això ens ho podríem permetre perquè plou sobre mullat: no emplenem caselles arbitràriament sinó amb valors que per definició són compatibles amb una solució.

Doncs bé: una alternativa a l'obtenció d'un SUDOKU-PROBLEMA partint de la prèvia disponibilitat de la SUDOKU-SOLUCIÓ (B, C) i a la creació del no-res (A), quedant-nos-en amb una part susceptible d'assegurar la restitució del conjunt, consistiria a recórrer a **OMPLE-SUDOKU** (protagonista de A, en la seva primera part) però no pas per actuar amb dos graus de llibertat (ordre d'afectació de les caselles i valor d'assignació) sinó per reintroduir aquesta solució casella a casella (només ens quedaria llibertat en l'ordre d'afectació) fins a completar-la. Com sabríem quan la solució única ja està determinada?: ho sabríem en acabar la introducció manual i completar-se automàticament l'emplenat per acció del dispositiu **ACTUALITZA-PLUS** (i més concretament, de **PLUS-N->P**) dins de la funció **ACTUALITZA**, circumstància que no obstaria per localitzar, rebobinant la filmació, el fotograma que recull aquest moment. Havent registrat l'ordre en què s'emplenaven les caselles i el nombre de valors d'assignació que oferia **MASCARA** en cada ocasió, el SUDOKU-PROBLEMA estaria constituït per les caselles emplenades manualment, excloses aquelles que haguessin format part de l'última seqüència ininterrompuda d'assignacions forçades (o sigui, fetes sobre màscares amb només un valor d'assignació disponible).

A diferència de les simplificacions aplicables quan el procediment s'utilitza per resoldre sudokus, si el volem per fabricar-ne, com a mètode alternatiu a l'exposat en els capítols precedents, només ens podrem permetre prescindir de **T+D<12**, però no alleugerir **SUD-MIN** acomodant-lo al cardinal d'un sudoku que encara no existeix. Menys encara suprimir-lo: necessitem quelcom que activi **POST**, per fer intervenir **RE-MEMBER** en la determinació dels valors d'assignació admissibles en cada casella, perquè en això (en el registre de les que anem emplenant, fins que només en quedin amb un únic valor possible) es fonamenta el sistema. Però hi ha un factor que ens permetrà transitar per **OMPLE-SUDOKU** en condicions més favorables que en la primera etapa de l'opció A, malgrat que **POST** s'activi prematurament: a diferència del que passava quan construïem una solució interactivament (**RE-MEMBER** havia d'explorar la viabilitat de tots els valors explícits de l'element de **LLL** associat a la casella, per incorporar-los a la màscara), quan la reconstruïm, molt probablement recorrent les caselles en un altre ordre però amb el valor d'assignació predeterminat, n'hi haurà prou a esbrinar si sols és viable aquest valor o n'hi ha algún altre, perquè de màscares buides ja no n'hi pot haver. Anomenant **M** el nombre de valors explícits de **(ELEMENT LLL)** en cada casella, l'execució de **MASCARA** serà més curta de mitjana:

- Perquè amb el valor predeterminat no cal perdre-hi temps executant **RE-MEMBER**: per definició ja sabem que és viable.
- Perquè, així que trobem un valor que sigui viable (el predeterminat no compta), ja podrem deturar la cerca i passar a una altra casella, a diferència de l'accés a **MASCARA** des d'**OMPLE-SUDOKU**, on també cada iteració **RE-MEMBER** acabava en trobar una solució compatible amb l'assignació del valor candidat a la casella (pitjor

era amb **RE+MEMBER** des d'**ANORMALS**, perquè calia localitzar totes les solucions) però l'havíem de repetir amb cadascun dels candidats, com hem dit fa un moment.
- Perquè, no podent-hi haver màscares buides, el cas més desfavorable no serà que **RE-MEMBER** hagi de fer **M** cerques llargues (explorar totes les combinacions entre les caselles buides) sinó que n'hagi de fer **(1- M)**. Fins i tot podríem fer que, si en les primeres **(1- M)** no s'hagués trobat cap valor viable, es passés a una altre casella sense parar-se a esbrinar quin era el **M**-èsim valor, perquè només podria ser el predeterminat, però no surt a compte afegir una condició més quan la confirmació rutinària del fet s'obté amb una comparació simple.
- Perquè ja no cal perdre el temps dibuixant màscares. La cerca de valors viables és purament virtual, i la manera d'emplenar la casella serà, com a **FES-PUBLIC**, únicament fent clic sobre les caselles. Això sí: aquí no ens podrem permetre de fer clic dos cops en la mateixa casella per revocar una assignació.

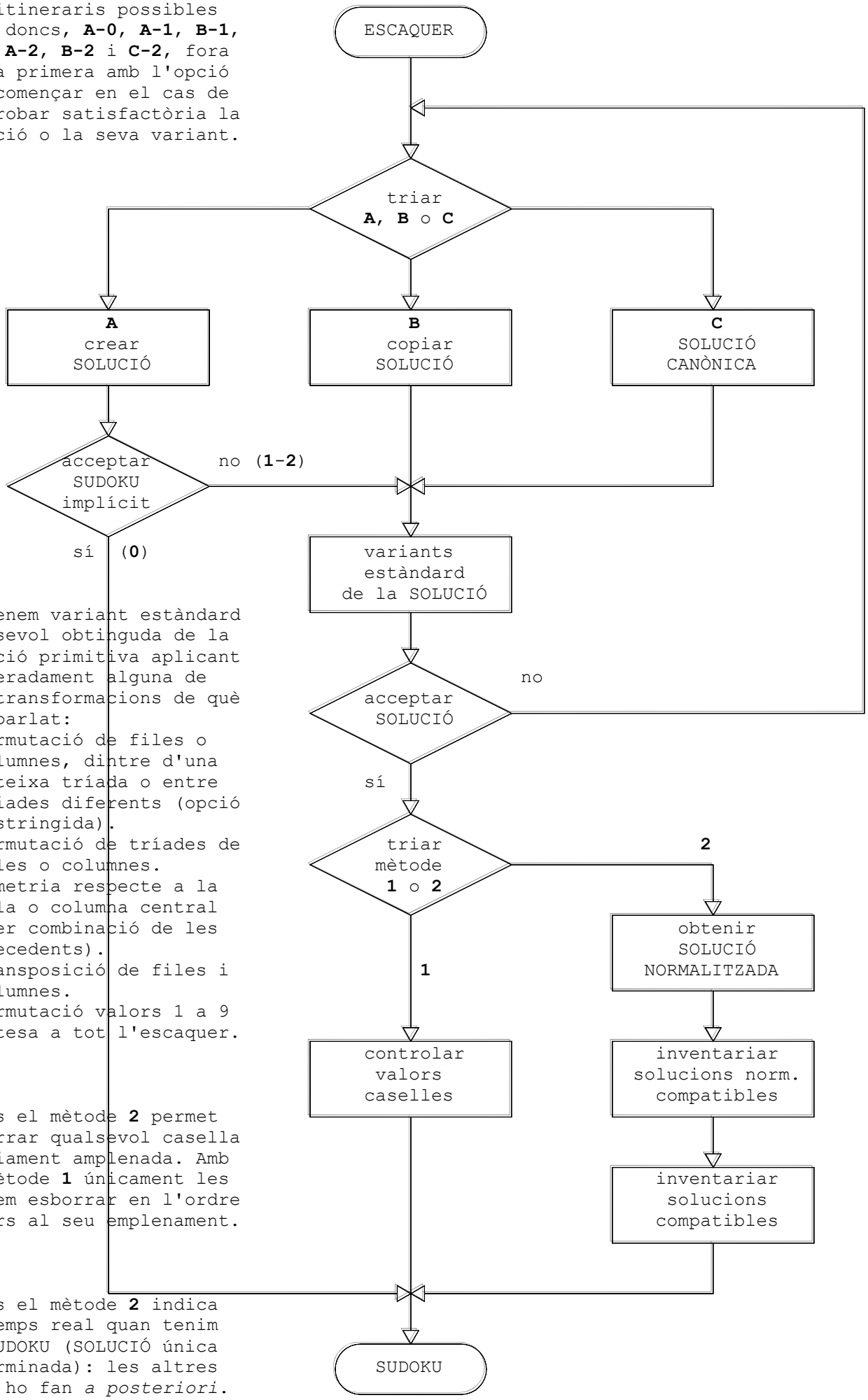
Si la decisió de fragmentar el procés de recompte de solucions en les tres etapes "PRE-FOTO", "FOTO 1-2" i "FOTO 2-2" del capítol precedent, desdoblant els elements separadors en dos, **(PERFIL-V*V)** + **(COMPAT-PERFILS)** d'una banda i **(ANORMALS)** de l'altra, representatius d'estratègies de resolució força diferents (en comptes de deixar-se de "solucions normalitzades" i d'altres històries, i limitar-se a dues etapes "PRE-FOTO" i "POST-FOTO" separades per un algorisme del tipus **(ANORMALS)**), va ser el típic pegat per falcar una construcció que quedava coixa a causa d'una badada inicial, però que unes succintes comprovacions van mostrar que bàsicament havia estat encertada (una vegada més, allò de "*sonó la flauta por casualidad*"), el dilema que se'ns obre ara, en suggerir un mètode alternatiu, és més compromés. Perquè el problema és saber què surt més a compte, a partir del moment en què **SUD-MIN** activarà **POST**: muntar guàrdia davant de l'ordinador, amatents als relativament soportables temps d'espera que s'aniran succeint, cada cop més breus, fins a la conclusió; o assumir la perspectiva de suportar dos períodes d'espera més llargs, quan **COMPAT-PERFILS** faci inventari de les solucions normalitzades compatibles i **ANORMALS** faci el mateix, d'una altra manera, amb les simplement compatibles, amb la possibilitat d'aprofitar les pauses per anar a dinar, a sopar... o a dormir.

L'autor ha realitzat algunes proves amb els dos mètodes i ha obtingut resultats dispars, pel que fa a la major eficiència d'un sobre l'altre, segons la SUDOKU-SOLUCIÓ utilitzada (fins i tot segons els diversos SUDOKU-PROBLEMA deduïbles d'una mateixa SUDOKU-SOLUCIÓ). Com també és conscient que el factor subjectiu (propòsit amb què s'utilitzi aquest programa, preferències personals, etc.) no serà alié a la valoració que se'n faci, ha decidit no prendre partit actuant selectivament, sinó oferir totes dues possibilitats i que l'usuari opti per la via més escaient. Així, si anomenem subopció "1" la que estem presentant com alternativa (però que encara no hem materialitzat en codi font) i subopció "2" la que procedeix dels capítols anteriors, podem considerar 6 itineraris d'execució: A-1, A-2, B-1, B-2, C-1 i C-2. Les peculiaritats de cadascun ja els anirem veient, perquè abans és prioritari aclarir possibles malentesos sobre el primer dels camins esmentats.

En què consistiria escollir l'opció A i després la subopció 1? Doncs en crear una SUDOKU-SOLUCIÓ des del no-res, mitjançant la funció **OMPLE-SUDOKU**, a partir de la qual compondríem el SUDOKU-PROBLEMA utilitzant un dispositiu que encara no s'ha concretat però que provisionalment hem presentat com molt semblant a **OMPLE-SUDOKU**: seria fàcil incloure en aquesta funció algunes sentències d'execució condicionada al lloc des d'on se la cridés, de manera que, des d'un punt de vista operatiu i esquematitzant les coses, podríem definir la via A-1 com **OMPLE-SUDOKU** executada dues vegades, amb variacions secundàries cada cop. Més que contradictori, això pot semblar absurd, per innecessari: si en la pàgina precedent suggeríem que, guardant ordenadament en una llista (o en llistes paral·leles, tant se val) les caselles que emplenàvem, el valor amb què les emplenàvem i el nombre de valors d'assignació viables (comptant amb l'utilitzat), i esborrant de la llista l'última subseqüència ininterrompuda d'assignacions forçades, assolida ja la SUDOKU-SOLUCIÓ, tindríem de retruc el SUDOKU-PROBLEMA corresponent a aquesta seqüència d'emplenament, ¿per què repetir l'operació? Es pot argumentar que, amb l'experiència del primer abordatge i amb la SUDOKU-SOLUCIÓ ja a la vista, la cerca d'un nou camí pot tenir interès, sobretot tenint en compte que la segona volta no comportarà pauses tan llargues. Ben cert, però això no justifica que aquesta última hagi de ser obligada: si convé plantar-se a la primera o passar a la segona ja ho decidirà l'usuari; pel que fa a nosaltres, quantes més possibilitats deixem al seu abast, millor.

A la pàgina següent hem dibuixat un diagrama que recull totes les possibilitats, incloent-hi l'opció a quedar-se amb el primer resultat, que anomenem drecera A-0.

Els itineraris possibles són, doncs, **A-0**, **A-1**, **B-1**, **C-1**, **A-2**, **B-2** i **C-2**, fora de la primera amb l'opció a recomençar en el cas de no trobar satisfactòria la solució o la seva variant.



Anomenem variant estàndard qualsevol obtinguda de la solució primitiva aplicant reiteradament alguna de les transformacions de què hem parlat:

- Permutació de files o columnes, dintre d'una mateixa tríada o entre tríades diferents (opció restringida).
- Permutació de tríades de files o columnes.
- Simetria respecte a la fila o columna central (per combinació de les precedents).
- Transposició de files i columnes.
- Permutació valors 1 a 9 estesa a tot l'escaquer.

Només el mètode **2** permet esborrar qualsevol casella prèviament amplenada. Amb el mètode **1** únicament les podrem esborrar en l'ordre invers al seu emplenament.

Només el mètode **2** indica en temps real quan tenim el SUDOKU (SOLUCIÓ única determinada): les altres vies ho fan *a posteriori*.

Anomenar "drecera" la via A-0, que teniu a l'esquerra, potser no sigui la metàfora més encertada, en la mesura que el procés A (crear SOLUCIÓ) és el mateix tant si la ruta és A-0 com A-1 o A-2, i l'únic que diferencia la primera de les altres dues és una decisió que es pren després d'aquest procés. Tal vegada fóra millor comparar-ho amb una elecció a dues voltes, però aquí la analogia grinyolaria pel fet que donar per bona la primera elecció o passar a una segona és discrecional del tot i no depèn pas de la majoria obtinguda, així que ens deixarem de romanços. Hem aprofitat els espais relativament lliures per inserir-hi alguns aclariments, de manera que, imprimint aquesta pàgina, puguem disposar d'una fitxa orientativa. Com que el penúltim d'aquests aclariments informa d'una realitat (el programa s'ha dissenyat amb aquestes característiques) però no se'n dóna cap justificació (per quina raó s'ha conformat precisament així), ara pot ser un bon moment per fer-ho.

L'ultim aclariment es fa ressò del que ja havíem anunciat i hem de veure amb més detall a les pàgines properes: que, en arribar a la solució única, el mètode 2 ens avisa de l'esdeveniment (tot i que podem seguir, conscients que estarem inflant el resultat, afegint-li redundància), però el mètode 1 i la drecera A-0 no tenen la capacitat d'informar-nos-en en temps real, fora del cas que, just després d'haver triat entre una oferta plural un valor d'assignació a una casella, s'esdevingués una reacció en cadena dintre de la funció **ACTUALITZA** (per l'activació reiterada de **PLUS**, impulsada des d'**ACTUALITZA-PLUS** -> **PLUS-N->P**) que automàticament ens acabés d'emplenar l'escaquer 9×9 (tot i així, la rapidesa del desenllaç ens privaria del coneixement precís del SUDOKU-PROBLEMA, tret que paral·lelament haguéssim anotat les caselles). Des d'aquesta perspectiva no costa massa de veure que pretendre fer com amb el mètode 2, en què un CLIC sobre qualsevol casella plena esborra el valor assignat, arrossegaria problemes gairebé insalvables (en el sentit de complicar les coses desproporcionadament en relació a allò que volíem aconseguir). Pensem, si no, què podria passar fent CLIC en una casella plena però que no fos l'última emplenada manualment (volem dir, no emplenada automàticament per **ACTUALITZA-PLUS**): suposant que la casella fos la **N**-èsima, si la següent, **(1+ N)**-èsima, també hagués estat emplenada manualment però sense opció quant al valor assignat (sols un valor admissible, és a dir $\mathbf{M_{n+1} = 1}$, circumstància que quedaria enregistrada per tal que aquesta casella fos ignorada en la recomposició final del SUDOKU-PROBLEMA), ¿n'hi hauria prou a esborrar l'element homòleg de la llista on s'enregistraven les caselles emplenades manualment (llista que, com hem avançat, hauria d'incloure el nombre **M** de valors d'assignació admissibles), en esborrar la primera? La resposta és que no, perquè si la seqüència d'emplenament hagués passat d'aquesta casella, anant directament a la segona, podria haver-se esdevingut que el nombre de valors d'assignació admissibles no fos $\mathbf{M_n = 1}$ sinó $\mathbf{M_n = 2}$ (ara tocaria anomenar-lo així i no $\mathbf{M_{n+1}}$), canvi que podria haver capgirat el resultat de la recomposició final.

Abans de seguir el fil del discurs, ens permetrem un incís perquè quan, d'aquí a no res, ens referim a ****9*9**** (i no a ****9**9****) i a ****V**V**** (i no a ****V*V****), la selecció no sembli arbitrària. Tot obeeix al "disseny de producció" de l'autor, que com haureu anat veient no és un exemple de previsió sinó més aviat d'anar fer front a troballes i esdeveniments imprevistos i, sobre la marxa, d'intentar salvar tol allò que mantenia alguna validesa, recomponent-ho i amanint-ho per conferir al producte un mínim de coherència. Per entendre-ho millor anem a la versió ultimada en el capítol precedent, en què només teníem un camí (que ara anomenem subopció o mètode 2) amb tres entrades alternatives (opcions A, B i C), i recordem que els dos escaquers 9×9 visibles en cadascuna de les finestres obertes en l'Espai Paper (**CVPORT** = **2** a l'esquerra i **CVPORT** = **3** a la dreta), de primer buits i després amb la SUDOKU-SOLUCIÓ repetida en ambdues bandes, no portaven cap títol durant el seu emplenament (gradual en les opcions A i B, instantani en la C) ni un cop complets. No era fins després d'haver llegit les possibilitats d'aplicar unes determinades permutacions de posició o de valor, que les finestres passaven a titular-se

SOLUCIÓ CREADA (A)
SOLUCIÒ COPIADA (B) VARIANT ESCOLLIDA
SOLUCIÓ CANÒNICA (C)

fins i tot abans d'aplicar les transformacions que fa poc hem anomenat estàndard. Els efectes de cada transformació es manifestaven exclusivament a la finestra de la dreta, i deixàvem a l'esquerra la SOLUCIÓ CREADA/COPIADA/CANÒNICA només com a referència. I, quan ja teníem una VARIANT ESCOLLIDA al nostre gust, el pas següent (obligat, en aquest camí únic) era la substitució, en la finestra de l'esquerra, de la SOLUCIÓ C/C/C per l'anomenada VARIANT ESCOLLIDA NORMALITZADA, derivada de la precedent amb permutacions entre requadres 9×3 i, dintre de cada requadre, amb permutacions entre files. Des d'aquest moment, els rètols

V. E. NORMALITZADA VARIANT ESCOLLIDA

es mantenien mentre durava el procés d'activació de caselles (l'assignació formal del valor predeterminat), activació que podíem efectuar des de qualsevol de les dues finestres (només calia el CLIC suplementari per passar d'una a l'altra) i que tenia confirmació visual (canviant de gris a blanc) des de les dues. Només quan, després de transitar per les tres etapes esquematitzades a la pàgina precedent (**POST** = **nil** sense cap recompte, **POST** = **T** amb recompte de solucions normalitzades compatibles de primer i amb recompte de totes les solucions compatibles després, etapes delimitades per l'execució de **COMPAT-PERFILS** i d'**ANORMALS**) ens quedava una única solució compatible, hauria convingut que la finestra de l'esquerra passés a visualitzar la VARIANT ESCOLLIDA amb tots els valors destacats en blanc (la V. E. NORMALITZADA era un simple recurs instrumental) i que els rètols canviessin a

SUDOKU-SOLUCIÓ SUDOKU-PROBLEMA

però, com tantes altres coses, aquesta floritura final no va tenir cabuda en la versió de què parlem, perquè a l'autor ja se l'havia encès la bombeta en relació a les vies addicionals de què tracta el capítol present. Abans de parlar d'aquestes altres vies, recordarem també que, mentre passàvem per les tres etapes esmentades (és a dir, mentre circulàvem pel tram específic del mètode 2) utilitzàvem certes dades per partida doble: de pseudomatrius 9×9 que emmagatzemaven la SUDOKU-SOLUCIÓ n'hi havia dos, ****9*9**** (el substrat gris de la VARIANT ESCOLLIDA NORMALITZADA, a l'esquerra) i ****9**9**** (el substrat gris de la VARIANT ESCOLLIDA, a la dreta), i d'aquelles que només emmagatzemaven els valors públics o visibles, per representar un SUDOKU-PROBLEMA, també n'hi havia dos, ****V*V**** i ****V**V**** (en color blanc, a la VARIANT ESCOLLIDA NORMALITZADA i a la VARIANT ESCOLLIDA respectivament).

Si tornem a l'etapa inicial, que podríem anomenar de "presolució", seguint l'opció B sembla raonable que, havent de limitar-nos a picar sobre el teclat virtual entre les dues finestres, els valors introduïts es visualitzin a les dues, encara que el cursor que avança de casella en casella només aparegui a l'esquerra, de la mateixa manera que, havent adoptat l'opció C, l'anomenada solució canònica s'imprimeix a dreta i esquerra: al cap i a la fi es tracta que, quan tinguem la SUDOKU-SOLUCIÓ provisional, el visualitzem a banda i banda per transformar-lo, si convé, en una VARIANT ESCOLLIDA diferenciada en la finestra de la dreta. Però, pel que fa al mètode A, ens preguntem si té gaire sentit de mantenir aquesta duplicitat, tenint present sobretot que, a més d'un mètode 1 alternatiu al 2, hi ha la drecera A-0. Diverses consideracions ens mouen a suprimir-la:

- Dintre de la etapa que podríem anomenar "postsolució", l'andròmina denominada VARIANT ESCOLLIDA NORMALITZADA només té sentit utilitzant el mètode 2 en què, per tal d'escurçar temps de processat, havíem decidit fraccionar la cerca de solucions compatibles amb la part pública de la VARIANT ESCOLLIDA en dos: només solucions normalitzades i solucions de tota mena. Amb el mètode 2 fins i tot es podria argumentar que aquesta opció estratègica i la virtualitat de la primera part del procés de cerca feien molt recomanable l'existència de ****9*9**** (versió normalitzada de la VARIANT ESCOLLIDA ****9**9****), pels petits avantatges que això comportava a l'hora de comparar reiteradament, però no que aquesta pseudomatriu hagués de tenir traducció gràfica, amb duplicitat no sols visual sinó operativa (possibilitat d'anar alternant els dos terrenys de joc). Si ens havíem molestat a introduir-la era imaginant que l'usuari podria orientar millor les decisions (en quines caselles convenia més fer CLIC) jugant en la finestra de l'esquerra, si més no abans que s'activés la variable **POST**, però això ja no té cap sentit des del moment que la premissa major (el *divide et vinces* a la cerca) deixa de ser aplicable al mètode 1, així que quan hagi optat per aquesta via l'usuari es limitarà a jugar en la finestra de la dreta. Ara bé, ¿què li aniria bé de veure mentrestant a la de l'esquerra, com a referència?:
  - De fet, en B-1 i C-1 podríem prescindir d'aquesta finestra, perquè l'interès visual ha passat a la dreta. Però, com que la desaparició d'un element gràfic que ens ha acompanyat des del començament (i que veurem de seguida el paper que té reservat a l'acabament) encara podria ser un factor de desorientació, optem per deixar-la com estava, com un inofensiu i gris *convidado de piedra*. No cal dir que els titulars seguiran sent

    SOLUCIÓ COPIADA/CANÒNICA VARIANT ESCOLLIDA
  - Per contra, en A-1 la finestra de l'esquerra podria tenir un paper secundari, certament, però d'una utilitat indiscutible: mantenir en blanc sobre gris les caselles del sudoku implícit en la fase presolució i anar canviant-les de posició i/o de valor amb les transformacions estàndard aplicades, per tal que, en crear el sudoku definitiu activant caselles a la finestra de la dreta, a l'usuari li fos més fàcil decidir en cada moment la conveniència de repetir els passos del primer o d'apartar-se'n (quan algú refusa de sortir del joc per la drecera A-0, hem de suposar que l'interessa treure de la SUDOKU-SOLUCIÓ un

SUDOKU-PROBLEMA diferent). Així que, tant si apliquem alguna transformació com si sortim del pas sense aplicar-ne cap, els títols que arrossegàvem

SOLUCIÓ CREADA VARIANT ESCOLLIDA

(en realitat, només el de l'esquerra) seran substituïts pels textos

V. E. & SUDOKU INICIAL VARIANT ESCOLLIDA

que serviran per il·lustrar la nova situació, però hem de precisar que aquesta configuració anirà lligada a A i no podrem discriminar la situació fins que la pregunta sobre el mètode a seguir (que l'hem preferit posterior a l'aplicació de les transformacions estàndard) ens situï just a la bifurcació de A-1 i A-2: pel primer camí, aquesta configuració ens acompanyarà fins el desenllaç final; pel segon, serà substituïda per la pròpia del mètode 2 (A-2, B-2 i C-2).

- Si ens hem de plantar a la primera volta (A-0), tampoc no té sentit duplicar la presentació mentre emplenem l'escaquer, tot i que al final convingui visualitzar per separat SUDOKU-PROBLEMA i SUDOKU-SOLUCIÓ (en aquest ordre o en el contrari), perquè ni tan sols se'ns presentarà l'opció a aplicar transformacions estàndard, única previsió que podria justificar-ho. És en atenció a aquest supòsit que, si no ens plantem i decidim anar a una segona volta, via A-1 o via A-2, esperarem a pronunciar-nos davant de la pregunta *Voleu seguir (S/N)? <S>:*, i només quan la resposta sigui positiva desdoblarem la presentació sobre dues finestres, encara anònimes. No serà fins després d'haver commutat a pantalla de text per llegir la relació de les transformacions aplicables, que apareixeran els títols

  SOLUCIÓ CREADA VARIANT ESCOLLIDA

  que esmentàvem unes línies enrera, de seguida substituïts pels textos

  V. E. & SUDOKU INICIAL VARIANT ESCOLLIDA

  Acceptada doncs la presentació unilateral en l'entrada per A, l'altra qüestió és quin costat triem per realitzar l'emplenament.

Si no ens plantem i seguim la trajectòria A-1, convé que la fase de postsolució es presenti amb algunes diferències formals respecte a la de presolució. Diferències secundàries, si voleu, però que sumades a la principal (en presolució disposem de dos graus de llibertat, que són l'ordre d'emplenament de les caselles i els valors d'assignació materialitzats en el teclat virtual, mentre que en postsolució només ens queda el primer, perquè els valors representats en gris són obligats) ajudin l'usuari a comprendre que la segona no és exactament una repetició de la primera: jugarem amb la lateralitat, usant la finestra de l'esquerra per a l'emplenament en fase de presolució i la finestra de la dreta en fase de postsolució. La disposició simètrica i l'adopció d'aquestes posicions en concret serviran diversos objectius:

- De la mateixa forma que mantenir la denominació VARIANT ESCOLLIDA en l'únic camp de joc que tenen en comú els mètodes 1 i 2 pot reforçar la percepció que entre les dues situacions hi ha el nexe de transitar per trams homòlegs (ens referim a creuar tots dos aquest territori anomenat postsolució) de carreteres d'idèntica categoria, recordar en el cas de l'itinerari A-1 que la fase presolució s'havia desenvolupat en el camp de l'esquerra ajudarà l'usuari a distingir una situació de l'altra (tot i les diferències esmentades mantenen certa semblança, sobre tot per la resposta -immediata o diferida- que es rep a cada nova casella emplenada, semblança que en l'itinerari A-2 queda força més diluïda) i, potser no del tot conscientment, a copsar-ne l'estructura, que és més complexa que no sembla pas a primer cop d'ull (encreuament entre una divisió horitzontal en dues etapes i una de vertical de tres opcions d'entrada i dos mètodes de sortida, amb porta falsa i tot) i a interioritzar-la per orientar-se millor en successives execucions del programa, encara que no s'hagi imprès l'esquema dibuixat tres pàgines enrera.
- Quant a les transformacions estàndard, pretenem que l'usuari associï la finestra de l'esquerra a la SOLUCIÓ creada en la primera volta amb l'opció A, que és el domini de ****9*9**** i ****V*V****, i la finestra de la dreta a la transformada VARIANT ESCOLLIDA, domini de ****9**9**** i ****V**V****, tret que utilitzem el mètode 2, en què l'associació és més artificiosa: tot i que la VARIANT ESCOLLIDA NORMALITZADA és un derivat de la VARIANT ESCOLLIDA, com que la SOLUCIÓ CREADA/COPIADA/CANÒNICA ja no hi pinta res, aquí substituirem el criteri de causalitat i, en comptes de considerar la genealogia real en muntarem una d'ideal per suposar que la VARIANT ESCOLLIDA és producte de la transformació de la VARIANT ESCOLLIDA NORMALITZADA, abstracció platònica que legitimarà les seves posicions a dreta i esquerra.

I, pel que fa a les variables en joc, la conclusió immediata és que, mentre que en A-0 o en fase de presolució de A-1 i A-2 (estem parlant d'un mateix procés) usarem valors que conformaran la SOLUCIÓ en primera volta, construint ****9*9**** (esquerra) casella a casella, igual com ho faríem amb l'opció B per altres mitjans i accedint al complex **MASCARA** + **ACTUALITZA** + **RE-MEMBER** amb aquesta pseudomatriu 9×9, en fase de postsolució de A-1, B-1 o C-1, en què ****9*9**** i la seva transformada ****9**9****

ja existeixen, l'accés a aquestes funcions es farà mitjançant ****V**V**** (dreta), construint de nou la mateixa pseudomatriu i, en el cas A-1, aprofitant la segona volta per emetre un vot diferent al de la primera (el "vot" no es refereix a uns valors que ja estan dats i beneïts sinó a l'ordre d'emplenament de les caselles). Conseqüents amb aquesta voluntat de diferenciar el trànsit per la drecera o pel camí ral mitjançant la simetría de posició, quan les trajectòries A-1, B-1 o C-1 hagin completat l'emplenament de caselles i fet la recomposició final, destacarem en blanc les caselles públiques del sudoku a la finestra de la dreta i reproduirem el seu contingut numèric a la finestra de l'esquerra (substituint el que s'hi veia fins aquest moment, que en A-1 diferia de B-1 i C-1, com assenyalàvem no fa gaire) però destacant en blanc totes les caselles, amb la titulació

SUDOKU-SOLUCIÓ SUDOKU-PROBLEMA

I, quan transitem per la drecera A-0, remarcarem en blanc les caselles públiques a l'esquerra (sobre la resta, gris) i la solució sencera a la dreta, amb els títols

SUDOKU-PROBLEMA SUDOKU-SOLUCIÓ

Aclarit això, proseguirem amb el que teníem entre mans.

Tot i acceptant la impossibilitat (o inconveniència pràctica, per ser més exactes) d'esborrar qualsevol casella amb valor assignat, sí que podríem baixar el llistó i conformar-nos a esborrar únicament l'última emplenada, ni que fos per evitar que, davant d'un CLIC accidental o en una casella equivocada, no hi hagués més solució que cancel·lar l'execució i tornar a començar (cosa ja prou enutjosa si la fallada s'ha produït a la fase d'emplenat de l'opció A, però exasperant si estàvem seguint el mètode 1, sobretot si hi portàvem una estona perquè transitàvem per l'itinerari A-1 o B-1). I, posats a fer, no costaria gens aconseguir que aquesta possibilitat fos permanent: que, després d'anul·lar l'última assignació, poguéssim anul·lar la penúltima i així successivament. Doncs bé: això ja és més assumible, sobretot si renunciem a implementar-ho en la fase inicial A i ho limitem al mètode 1, però tot i així comportarà més canvis que no semblava a primer cop d'ull, perquè potser el vessant gràfic de l'anul·lació es podria resoldre associant a l'emplenament manual o automàtic de les caselles un dispositiu de marcatge amb **(command "DESHACER" "M")** que es podria anar revocant, sempre des del final (com correspon a una estructura de pila), amb **(command "DESHACER" "R")**, però l'actualització interna no s'acabarà amb l'afectació de les pseudomatrius 9×9 ****V**V**** i **LLL** sinó que caldrà intervenir igualment a la llista-registre de què hem parlat, que anomenarem **PPM**. A més,

- Contra allò que havíem previst inicialment, caldrà incloure en **PPM** les caselles emplenades automàticament des d'**ACTUALITZA-PLUS**, no únicament per poder-nos-les saltar en la recomposició final i quedar-nos estrictament amb les que determinen la SUDOKU-SOLUCIÓ sinó, en general, per evitar solucions de continuïtat a l'hora de reculer, casella a casella, pel camí recorregut (l'alternativa a la inclusió dels forats tapats en **PPM** passaria inexorablement per la creació d'una llista específica, solució poc recomanable perquè encara ens ho complicaria més tot). En conseqüència, per tal d'actuar com convingui en cadascuna de les situacions esmentades, haurem d'ampliar la diferenciació del valor **M** que, conjuntament amb les coordenades de la casella, ha de constituir l'element de la llista-registre (en realitat utilitzarem dos punts en comptes d'un, i per això l'anomenem **PPM** i no **PM**): en comptes de limitar-nos als valors **2** (en el cas que la casella tingui dos o més valors viables) i **1** (en el cas que només el valor predeterminat sigui viable), haurem d'incorporar el valor **0** (caselles en què, com les precedents, només el valor predeterminat és viable, però que són d'emplenament automàtic).

- Mentre anem emplenant caselles, sigui a la fase de presolució de A o a la fase de postsolució seguint el mètode 1, la funció **MASCARA** ens haurà d'informar dels valors admissibles per a la casella que anem a omplir, amb una inspecció simple a la pseudomatriu **LLL** (**(member N (ELEMENT LLL))**, si encara era **POST** = **nil**) o més prospectiva (esquemàticament, **(RE-MEMBER xxx LLL)**, si ja és **POST** = **T**, on amb **xxx** volem representar ****9*9**** a la fase presolució ja resolta, i ****V**V**** a la fase postsolució que estem perfilant). Només per això, ja es comprèn que cal disposar de **LLL** i anar-la actualitzant amb la funció **ACTUALITZA**, la necessitat de la qual ja havíem acceptat implícitament; la de ****V**V**** podria haver anat a càrrec de la funció **CLIC** (igual que amb el mètode 2, interessa que la confirmació visual d'haver activat una casella sigui immediata, perquè l'usuari no s'impacienti i repeteixi el clic) però la inhibició d'**ACTUALITZA** duria més feina, així que **CLIC** il·luminarà la casella (com **TECLA-M** situa el visor a **FES-SOLUCIO**) i **ACTUALITZA** actualitzarà ****V**V****. La novetat es presenta en considerar la revocació de les caselles emplenades: per més que la restringim a una regressió des de l'última,

caldrà guardar els elements de **LLL** corresponents a cada casella; si no, havent arribat a l'última que volíem esborrar, no podríem accedir a la funció **MASCARA** per saber quins altres valors podien haver-se-li assignat. I, si esteu pensant que n'hi hauria prou a emmagatzemar les màscares (una llista de pseudomatrius 3×3 faria menys embalum que una formada per pseudomatrius 9×9×9), us hauríem de puntualitzar que, si l'usuari ha decidit fer marxa enrera fins a una casella determinada, de segur que serà per refer el camí esborrat amb valors nous o per seguir-ne un altre, de manera que les màscares emmagatzemades ens servirien de ben poc i ens veuríem obligats a començar de zero **LLL** (l'estat corresponent a aquesta casella) per tal d'anar-la actualitzant a cada pas endavant: fet i fet, més val guardar les pseudomatrius **LLL** des d'un principi, perquè les temptatives d'estalvi (per exemple, començar-les a guardar a partir del moment d'activació de **POST**) encara portarien més complicacions. D'aquesta llista en direm **LLL/LLL**.

- Dels dos paràgrafs precedents se'n dedueix fàcilment la necessitat d'ampliar **LLL/LLL** a les caselles emplenades automàticament. Ara bé, si **PLUS-N->P** no només s'ha d'ampliar amb assignacions per actualitzar **PPM**, com admetíem implícitament, sinó que cal fer el mateix amb **LLL** i **LLL/LLL**, per evitar duplicitats a l'hora d'actualitzar **LLL** convindrà segregar el dispositiu que fins ara teníem integrat en **ACTUALITZA** i independitzar-lo com a funció, sota la denominació d'**ACT-LLL**.

Com que l'esquema inicial ha variat sensiblement, ens permetrem avançar que a la funció **FES-PUBLIC**, que fins ara concentrava la fase de postsolució, l'anomenarem des d'ara **FES-SUDOKU** i la desdoblarem en **FES-SUDOKU-1** i **FES-SUDOKU-2**, segons que respongui a un o altre mètode, i que la funció **OMPLE-SUDOKU** on s'aplegaven les opcions no trivials de la fase de presolució (A i B) passarà a dir-se **FES-SOLUCIO**. Abans de materialitzar les incorporacions i canvis descrits en aquest capítol en una versió definitiva del codi, però, haurem de retornar al precedent per recosir un trau que hi havia quedat i que, en principi, sembla afectar només el mètode 2. En concret, cal situar-se al final de la primera pàgina (p. 389) d'aquell capítol, just on, tornant al sudoku "difícil" de LA VANGUARDIA del 25-02-08, confirmàvem la sospita que haver arribat a una única solució normalitzada compatible no era ben bé haver arribat al cap del carrer. Quan encara quedava per activar el valor **4** de la casella 5,3 a la V. E. NORMALITZADA (5,5 a la VARIANT ESCOLLIDA) el missatge "EL SUDOKU JA TÉ UNA SOLUCIÓ NORMALITZADA ÚNICA" ens feia reflexionar sobre això. A la dreta tornem a reproduir el quadre on, en aquesta situació, es representa a cada casella el nombre de solucions compatibles que quedarien activant-la, però ara es representa sobre la VARIANT ESCOLLIDA. Repetim que, a més de la 5,5 esmentada, hi ha la casella 1,6, que en omplir-la (també amb un **4**) assoliria la SOLUCIÓ ÚNICA (**001**), i que activant les caselles 9,2, 9,3 o 7,9 no variaria el nombre de solucions compatibles (**593**): emplenar-les només afegeix redundància.

---	291	027	024	004	---	**593**	---	---
187	221	188	422	176	220	---	---	247
217	---	034	---	027	372	430	100	069
001	---	355	002	047	---	197	201	204
008	067	---	---	**001**	---	---	038	355
204	247	058	---	043	090	449	---	055
238	212	204	211	148	---	086	---	**593**
186	224	---	024	067	027	020	478	**593**
---	291	042	---	002	152	148	202	---

Això vol dir que un usuari que només hagués tingut accés a la solució del diari de l'endemà (26-02-08), tingués ganes de buscar un SUDOKU-PROBLEMA a partir d'aquesta SUDOKU-SOLUCIÓ publicada, hagués anat activant caselles coincidint casualment amb les del sudoku publicat el dia abans i es trobés en el punt que consideràvem, però se li estronqués de cop la comunicació telepàtica amb Michael Mephan, pare de la criatura, i, abans d'arribar al final activant la casella 5,5, li donés per fer-ho amb les caselles 9,2, 9,3 i 7,9, ¿seria víctima indefensa d'un programa traïdor que, en dir *EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA. Omplint més caselles el fareu més fàcil (<Intro> o < > per acabar)*, sembla suggerir que si no s'omplen més caselles el sudoku serà estricte i no tindrà redundància? Doncs no. Seria exagerat parlar d'indefensió, perquè els missatges que haurà anat rebent no podien ser més clars: per limitar-nos als missatges de l'àrea de text, si després d'activar la casella 4,4 (**6**) li diuen *Teniu 1 solució normalitzada compatible amb V. E. NORMALITZADA. Hi ha 592 solucions més, no normalitzades però també compatibles, així que heu d'omplir més caselles*, i després de cadascuna de les activacions inútils 9,2 (**4**), 9,3 (**5**) i 7,9 (**3**) apareixen sengles informacions, totes dient *Amb 593 solucions heu d'omplir més caselles*, abans d'arribar al missatge *EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA...* amb l'activació de 5,5 (**4**), l'usuari només ha d'interpretar correctament aquests missatges i fer-se enrera desactivant les tres caselles, tret que li vagi bé inflar el sudoku; si més no, sempre podrà recórrer al mètode d'assaig i error.

La llàstima és que en l'etapa "FOTO 1-2" les conclusions no siguin tan clares com a "FOTO 2-2": si, en activar una casella, el decrement de solucions normalitzades compatibles és nul, en principi això no pressuposa res: el valor d'aquesta casella és present en totes les solucions normalitzades compatibles amb les caselles que ja eren públiques, però això no implica que també hagi de figurar en les solucions compatibles no normalitzades. Una manera de tenir-ne la certesa seria adoptar les variants que en el capítol precedent anomenàvem **alternatives 1** i **2**, en què **POST** posava en marxa una versió modificada de **COMPAT-PERFILS** en un cas o **ANORMALS** en l'altra, per inventariar directament totes les solucions compatibles. Però, com que en termes d'ocupació de memòria i durada del procés ja hem comprovat l'escassa operativitat d'aquestes variants, si realment ens interessés tant saber si darrera de la no afectació del conjunt de solucions compatibles normalitzades hi ha hagut o no minva en el de solucions compatibles (per haver minvat o no el de solucions compatibles no normalitzades), sempre podríem recórrer un procediment poc elegant però eficaç: prendre nota d'aquelles caselles l'activació de les quals no comporta alteració en el nombre de solucions compatibles normalitzades i, quan només ens en quedés una, de solució compatible normalitzada (**FOTO2**), emprendre una marxa enrera desactivant caselles fins a quedar-nos amb aquestes de pes específic desconegut; ara sí que, desfent camí en el domini "FOTO 2-2", els missatges faran referència al total de solucions compatibles i, cada vegada que resti com a última emplenada alguna de les caselles irrellevants quant a solucions compatibles normalitzades, resoldrem el misteri comparant aquest total amb la quantia anotada abans i podrem optar entre mantenir-la emplenada o desactivar-la per transitar altres viaranys.

Però considerem una altra possibilitat: si, abans de reblar el sudoku amb 5,5 (**4**), activem la casella 4,9 (**4**), el missatge *Amb 24 solucions heu d'omplir més caselles* no aixecarà cap sospita; però si, un cop assolida la solució única, decidim de desactivar-la i confirmem l'acció malgrat els advertiments, al cap d'una espera breu (ara tenim 23 caselles públiques i dintre d'**ANORMALS** no es produiran tantes recursions de **RE+MEMBER**) tornarà a aparèixer l'anunci triomfal *EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA...* ¿Això vol dir que la casella 4,9 (**4**) era tan accessòria com 9,2 (**4**), 9,3 (**5**) i 7,9 (**3**)? Doncs sí, i aquesta vegada ningú no ho podia preveure... tret que hagués detectat que, amb el valor **4** a les caselles 6,3, 5,5 i 7,8, i amb la casella 4,7 també ocupada, el **4** a la 4,9 ja estava cantat, perquè aquesta és una de les configuracions que **ACTUALITZA-PLUS** < **ACTUALITZA** detecta i que **PLUS-N->P** remata, emplenant la casella (i, des del mètode 1, incloent-la a la llista **PPM** amb la marca **0**, com dèiem dues pàgines enrera). Ara bé, aquesta actuació aparentment contradictòria d'emplenar automàticament unes caselles que després s'eliminaran (les emplenem per completar quan abans la solució, estalviant-li feina a l'usuari, i les tornem a buidar per llevar-li redundància al sudoku) és exclusiva de l'opció A (presolució) i del mètode 1 (postsolució), cosa que podeu comprovar si aneu a la primera, reproduïu la seqüència d'emplenament descrita i després de 5,5 (**4**) ompliu algunes caselles més amb l'únic valor admissible que us mostrarà la màscara 1... 9 fins que, entre emplenaments manuals i automàtics, es completi la solució (talment allò que us suggeríem a l'inici del capítol per resoldre sudokus, però en aquest cas sense necessitat de desactivar **T+D<12** ni de trucar **SUD-MIN**); un cop assolida la solució, digueu que voleu seguir i, sense perdre el temps transformant-la amb **CANON->VARIANT** (veureu com a l'esquerra apareix la V. E. & SUDOKU INICIAL, amb la depuració de les caselles emplenades automàticament, entre d'altres), adopteu el mètode 1 i reproduïu un cop més la seqüència d'emplenament, limitant-vos aquesta vegada a activar les caselles (que ja tenen el valor potencial predefinit): veureu com, en ambdues ocasions, si en emplenar 9,1 (**3**) ja teníem ocupades 7,8 (**4**) i 8,8 (**5**), les caselles 9,2 (**4**) i 9,3 (**5**) s'emplenen automàticament (o, si no, ho fan en emplenar-se les primeres); en emplenar 4,7 (**3**) passa el mateix amb 7,9 (**3**) i, en emplenar 5,5 (**4**), ho fa 4,9 (**4**). Disortadament, com que el mètode 2 no es basa a desfer el camí que ens ha dut al cim (les 9×9 caselles activades) fins un punt a partir del qual ja no presenti bifurcacions, no li cal cap dispositiu tapa-forats, de l'estil d'**ACTUALITZA-PLUS**, avançant-se a l'acció de l'usuari... però ara ens adonem que hauria estat un bon ajut per no anar tan a les palpentes en l'activació de caselles, si més no en aquells casos (casella 4,9) en què la informació que ens va subministrant el programa no és tan significativa com per posar-nos en alerta (l'activació de les caselles 9,2, 9,3 o 7,9, en totes les quals el decrement del nombre de solucions compatibles es nul, són casos límit) i fer-nos reaccionar de forma immediata, duent-nos a desactivar l'ultima casella per provar-ne una altra.

De tota manera, abans d'implementar un dispositiu així en un disseny del tot aliè, com el de l'anunciada **FES-SUDOKU-2** (l'antiga funció **FES-PUBLIC**), convindria estar segurs que no hi haurà cap altra porta per on se'ns pogués colar la redundància... tot i que potser estem confonent dos conceptes que estan emparentats però no són equivalents: d'una banda, podem parlar de redundància *stricto sensu* quan, assolida la SUDOKU-SOLUCIÓ, seguim emplenant caselles (eventualitat sobre la qual ja se'ns avisa amb allò de *EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA. Omplint més caselles el fareu més fàcil*) i també quan, en fase "FOTO 2-2", veiem que el decrement en el nombre total de solucions compatibles es nul; també ho seria quan, en fase "FOTO 1-2", un decrement nul en el nombre de solucions compatibles normalitzades es revelés com a decrement nul en el total de solucions compatibles (per confirmar-ho caldria allò que suggeríem a la pàgina precedent: obtenir **FOTO2** i retrocedir al mateix punt); de l'altra, quan parlàvem d'activar la casella 4,9 (**4**) (*Amb 24 solucions ...*) en comptes d'activar 5,5 (**4**) (*EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA*), just després del missatge *Hi ha 592 solucions més, no normalitzades però també compatibles*, de fet no hauríem de parlar de redundància sinó d'opció poc eficient, si allò que volem és assolir el SUDOKU-PROBLEMA amb el menor nombre possible de caselles emplenades, a partir de la SUDOKU-SOLUCIÓ adoptada. Perquè no sempre és això el que volen els creadors de sudokus: deixant de banda que la correlació entre nombre de caselles lliures i grau de dificultat del sudoku és molt feble, moltes vegades l'objectiu prioritari no és la mínima ocupació per determinar una SUDOKU-SOLUCIÓ (si només comptés això, invariablement hauríem d'anar a petar als 47.793 sudokus bàsics de Gordon Royle i els seus derivats, amb només 17 caselles plenes), sinó aconseguir determinades ordenacions geomètriques de les ocupacions (simetria axial respecte a la 5ª fila, columna o respecte a les dues alhora, o simetria central respecte a la posició 5,5) o determinades figures (aproximacions a quadrats, octògons, a creus gregues, potençades o gammades, a més o menys reixides falces i martells, com les utilitzades cap al final del capítol 4 i començament del 5), tot això sense tenir en compte els valors d'assignació o jugant també amb ells (per reforçar l'efecte visual de les simetries amb separacions parells-senars o procurar que les caselles ocupades que definien alineacions tinguessin els valors ordenats, joc que havíem incorporat en el disseny de símbols nazis i estalinistes d'aquells dos capítols).

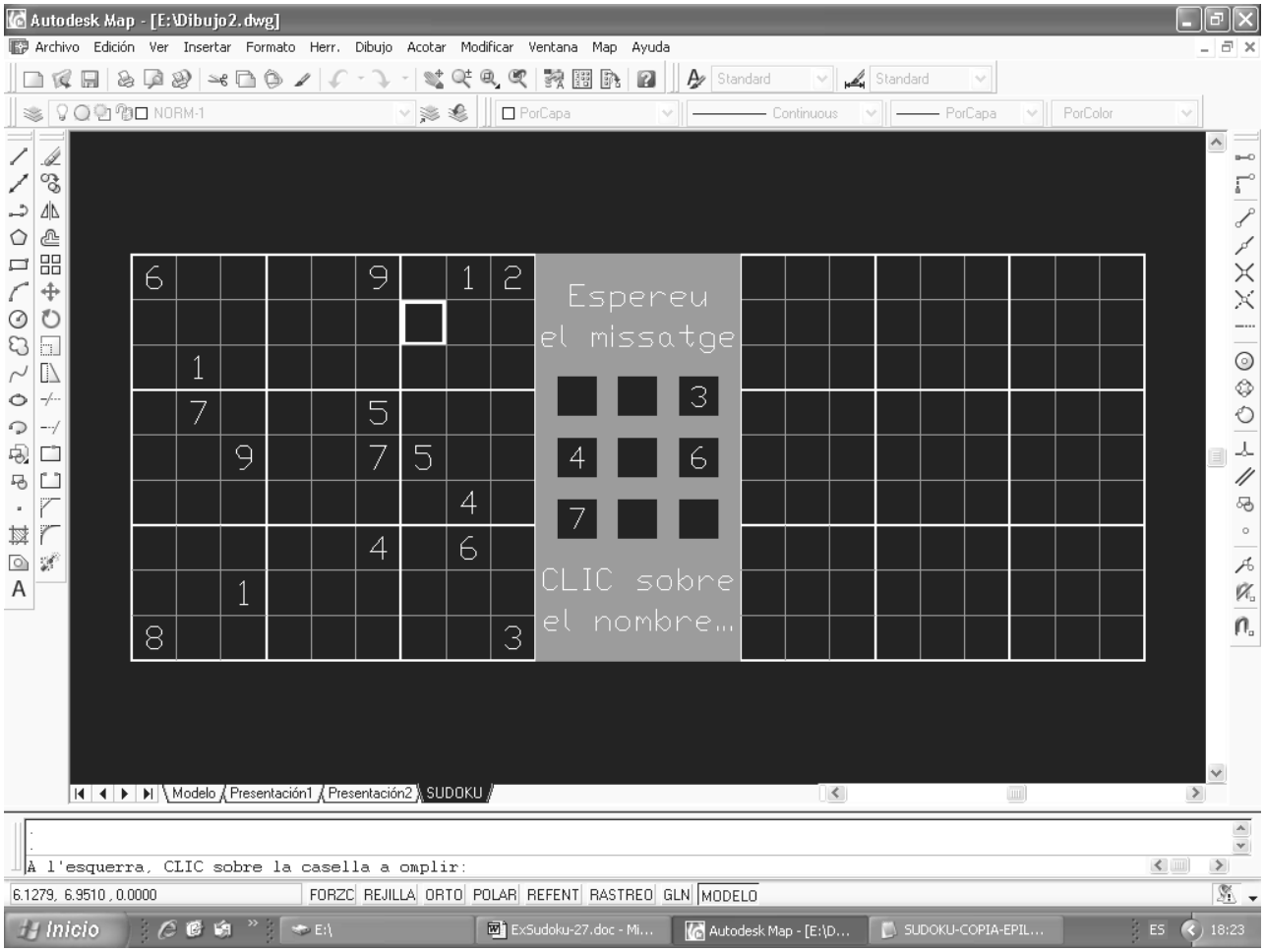
Tot això ens hauria de fer recapacitar, adonar-nos que algunes de les intencions manifestades són contradictòries i acabar admetent que, malgrat no haver-hi hagut cap propòsit de partença en la línia de crear dos mètodes orientats a dos tipus diferents d'usuari (recordeu que abans del present capitol només en consideràvem un, de mètode), l'atzar ens ha dut a certa idoneïtat de l'un i de l'altre quant a possibilitats del joc: tot i la impressió provisional que el mètode 1 és superior en velocitat, aquest pot no satisfer les expectatives d'aquell usuari-creador que prioritzi la capacitat de reproduir una forma prefixada, perquè l'actuació de la funció **ACTUALITZA-PLUS** no sempre li permetrà d'emplenar les caselles al seu gust, per tal d'arrodonir la imatge visual concebuda (a més de l'acció tapaforats, que proscriurà determinades caselles com a integrants del SUDOKU-PROBLEMA, cal tenir en compte les caselles en què **RE-MEMBER** < **MASCARA** accepta un únic valor candidat, no intervingut per **ACTUALITZA-PLUS** com a forat a tapar però que s'ha quedat sol); per contra, i precisament per la seva incapacitat de copsar quan l'activació d'una casella és innecessària (perquè ja figura en totes les solucions compatibles amb la part pública del sudoku, i en particular en la que quedi com a solució única), el mètode 2 permetrà a l'usuari modelar l'escaquer 9×9 com li vingui de gust (aixo sí, obligant-lo a suportar de tant en tant unes esperes de jutjat de guàrdia), i és per això que trobem preferible de mantenir aquesta incapacitat. Una altra cosa seria que SUDÒKULUM aspirés a assegurar a l'usuari la consecució del(s) millor(s) SUDOKU(s)-PLOBLEMA possible(s) a partir d'una SUDOKU-SOLUCIÓ determinada (assumint que per "millor" entenem el més buit) o, si més no, a guiar-lo en aquesta empresa, però el fet que un tal propòsit no hagi figurat mai entre els plans de l'autor no obsta perquè el repte no pugui ser recollit per algun lector d'esperit inquiet...

Així doncs, mantindrem aquest mètode 2 en la formulació actual, en el benentès que ningú no està obligat a utilitzar-lo: haver assenyalat que va bé per temptejar les possibilitats d'una SUDOKU-SOLUCIÓ de cara a certs tocs formalistes d'última hora, no exclou que, fins i tot per atendre aquests sobrecondicionaments, l'ús reiterat de l'opció A pot ser més curt; tot depèn de l'abast de les esmenes que calgui fer. Per exemple, si algun torradellonses no considerés prou satisfactòries les creus gammades del capítol 4, en creure que les 8 caselles emplenades a les cantonades (1,1, 1,2, 1,9, 2,9, 9,9, 9,8, 9,1 i 8,1, per arribar a una solució única que les 17 caselles de l'esvàstica pròpiament dita no determinen) reforcen la suggestió de gir d'aquesta figura, i que seria preferible que el suplement definís un marc més

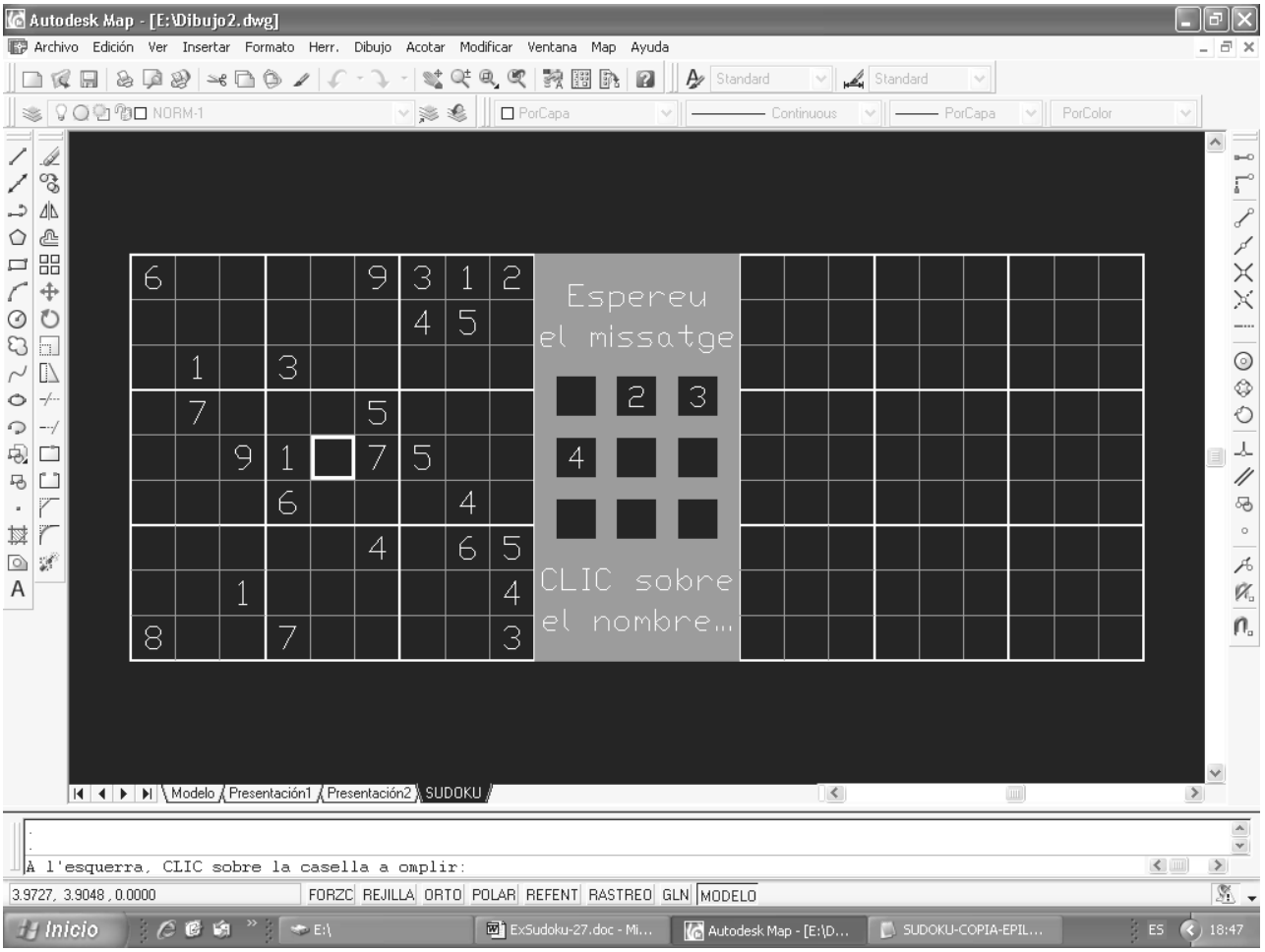
estàtic, com a la representació de l'esquerra (caselles 2,1, 1,2, 1,8, 2,9, 8,9, 9,8, 9,2 i 8,1), no li serviria la mateixa SUDOKU-SOLUCIÓ, sinó que hauria d'anar provant amb l'opció A fins a veure que amb la composició 1-2-3-4-5 del requadre 3×3 central no hi ha res a fer i cal canviar a 1-2-4-6-8 (si és que vol seguir amb la continuïtat numèrica de les 4 últimes xifres del quintet en els braços) per tal de crear els petits xamfrans; ampliar-los a tres caselles (configuració més a prop del cercle que emmarca l'esvàstica en la simbologia nazi) ja comportaria passar de 25 a 29 caselles plenes, incorrent en redundància. Únicament si aquest sacrifici a l'estètica no importa, es pot recórrer al mètode 2 per mirar de millorar l'aspecte d'una solució obtinguda mitjançant l'opció A: al mig, el martell encreuat sobre la falç que vèiem a l'inici del capítol 5 i que tenia el cap mal contrapesat respecte al mànec, es por equililibrar a costa de fer-lo exageradament gran i d'anar a una solució redundant, emplenant la casella 5,7 (**7**); a la dreta, tornant a l'Alemanya hitleriana (perquè la intenció, segurament no molt reixida, era dibuixar les esses rúniques de les SS) però jugant la carta oposada a l'exemple de l'esquerra, en la línia d'emplenar caselles més enllà d'una SUDOKU-SOLUCIÓ estricta, hem d'explicar que les dues esses s'enduien 18 caselles plenes (en les dues, l'ordenació completa de l'1 al 9) i que calia afegir-ne 8 més per assolir la SUDOKU-SOLUCIÓ, 6 partides en dues alineacions separadores i 2 en les caselles oposades 1,9 i 9,1, però que aquestes últimes quedaven al marge de la tendència a estructurar-ho visualment tot en la direcció de la diagonal principal, raó per la qual es va optar per anar a un total de 30 caselles ocupades, afegint les caselles 1,8 i 2,9 a la 1,9 de l'angle superior esquerre i les caselles 9,2 i 8,1 a la 9,1 de l'angle inferior dret. Aquí encara podríem efectuar una distinció entre aquestes dues últimes representacions, en funció del que l'usuari amb pretensions de dissenyador gràfic volgués fer a la vista de la SUDOKU-SOLUCIÓ estricta: en la del mig ja la tenia amb les 26 caselles d'abans, però la visió del martell estrafet li demanava reequilibrar-lo afegint la casella 5,7, acció per a la qual n'hi havia prou a retocar manualment el dibuix, si l'execució de SUDÒKULUM havia arribat al seu fi; en la de la dreta, en canvi, suposarem que la primera intenció de l'usuari no era sumar els parells de caselles 1,8 (**5**) i 2,9 (**3**) a 1,9 (**6**) ni sumar el parell 8,1 (**7**) i 9,2 (**5**) a 9,1 (**4**), sinó substituir-les quedant-se amb una solució redundant de només 28 caselles, perquè aquests parells contribuïen millor a reforçar visualment el predomini exclusiu de la direcció diagonal, però en desactivar 1,9 i 9,1, tornant a veure-se-les amb 33 solucions compatibles, i tot seguit activar 1,8, 2,9, 8,1 i 9,2, contra totes les previsions es va trobar que no havia remuntat prou recuperant una solució única, sinó que encara subsistien 16 solucions compatibles, adversitat que el va dur a conformar-se amb una solució formal de compromís, restaurant 1,9 i 9,1 (val a dir que, sense el concurs del mètode 2, el procés encara hauria resultat més feixuc).

4 **7** 2 **6** 8 5 1 9 **3**	9 **5** 1 **4** 2 3 6 8 7	6 **8** 3 9 7 **1** **5** 2 4	**9** 6 8 3 5 7 2 1 **4**	2 3 **1** 6 4 8 **5 7** 9	4 7 5 **2** 1 9 8 **3** 6	**6 3** 2 **5** 1 9 4 8 **7**	5 **9** 7 **8** 6 4 1 2 **3**	4 **1** 8 **2** 3 7 6 5 **9**
9 5 7 **2** 6 8 3 **1** 4	**2** 3 **4** 5 **1** 9 **8** 7 **6**	1 **6** 8 3 4 **7** 2 5 9	8 **3 2** 6 7 **1** 4 9 5	**9** 6 5 3 **8** 4 1 2 **7**	7 **4** 1 9 **5** 2 **3 6** 8	7 **6 5** 9 4 **3** 8 **2** 1	**4** 3 1 2 5 8 9 7 **6**	9 **8** 2 **7** 6 1 **5 4** 3
8 2 **9** **5** 3 1 7 **4** 6	3 4 5 7 6 **8** 1 **9** 2	**7** 1 6 4 9 **2** 8 **3** 5	5 8 **3** 7 **2** 9 **1** 4 6	7 1 **2** **4 5 6** 8 9 3	**6** 9 4 1 **8** 3 5 2 **7**	**1** 5 4 3 7 **8** 2 **9** 6	**7** 8 9 6 4 **2** 3 **1** 5	**3** 2 6 1 9 **5** 8 **7 4**

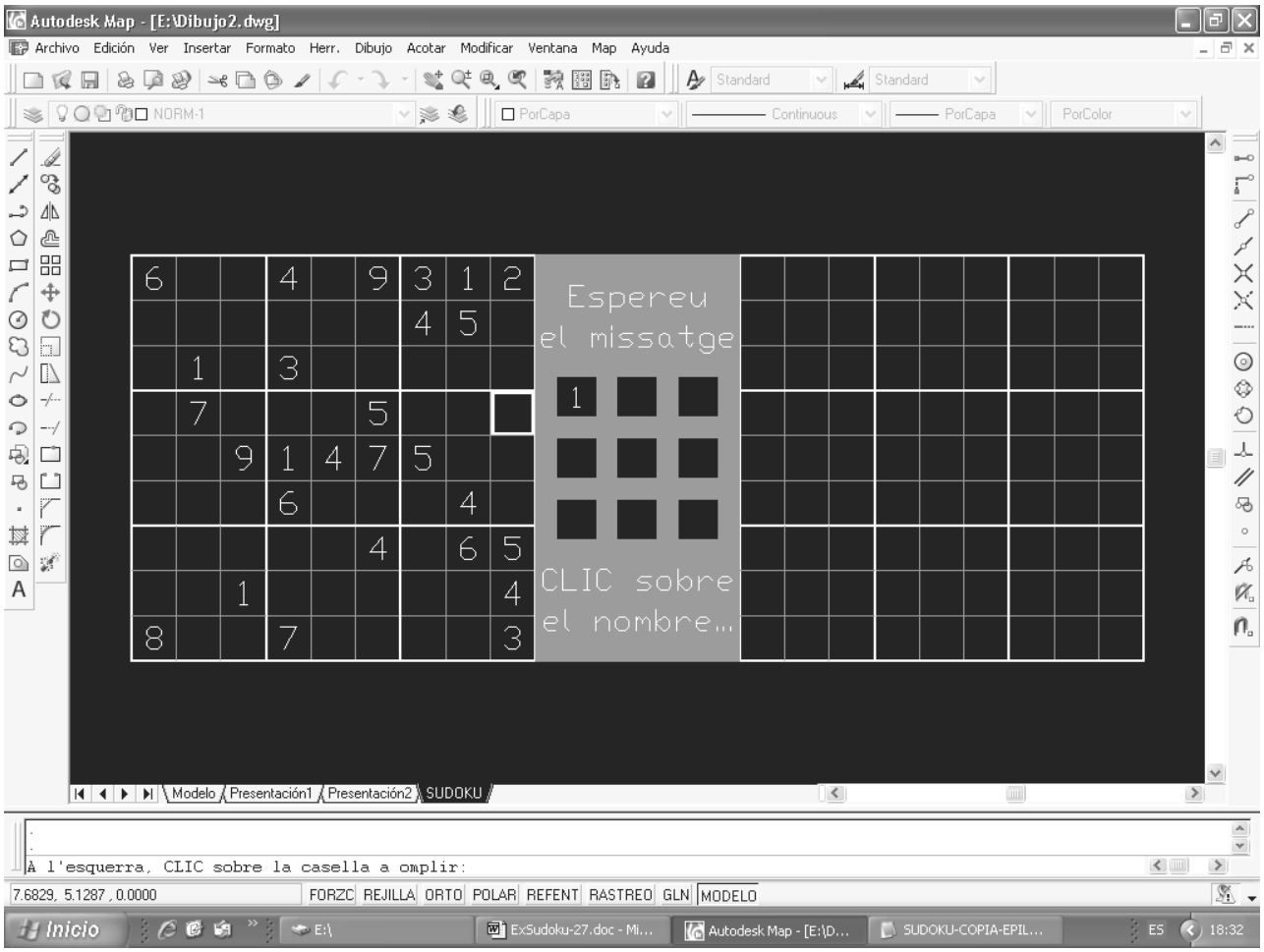
Deixem provisionalment la defensa del mètode 2, per presentar la traducció a codi de tot quan hem vingut proposant en aquest capítol, i diem provisionalment perquè aquest mètode, fonamentalment per una funció **SUD-MIN** que encara és massa rígida i que dispara **COMPAT-PERFILS** massa prematurament, segueix sent d'aplicació limitada a uns pocs casos, i encara si abans la seqüència inicial d'emplenaments ha estat ben meditada. De fet, fèiem trampa quan en el paràgraf precedent presumíem d'haver temptejat les possibilitats de modificar els angles superior esquerre i inferior dret del sudoku SS, fins i tot quantificant les solucions compatibles amb la part pública de la SUDOKU-SOLUCIÓ adoptada, en desactivar les caselles situades allà: en realitat, havíem hagut de manipular **SUD-MIN** perquè tot el que s'explicava fos realitzable, en la línia d'allò que, més raonadament, es proposarà el el capítol final *Consells pràctics per evitar una executio precox*; ni les esmenes introduïdes més endavant, en un apèndix en negreta al final del capítol present (i anunciat al final del capítol *Etapa prèvia: crear una solució, V*) aconseguiran dotar al mètode 2 d'una velocitat de processat mínimament acceptable.



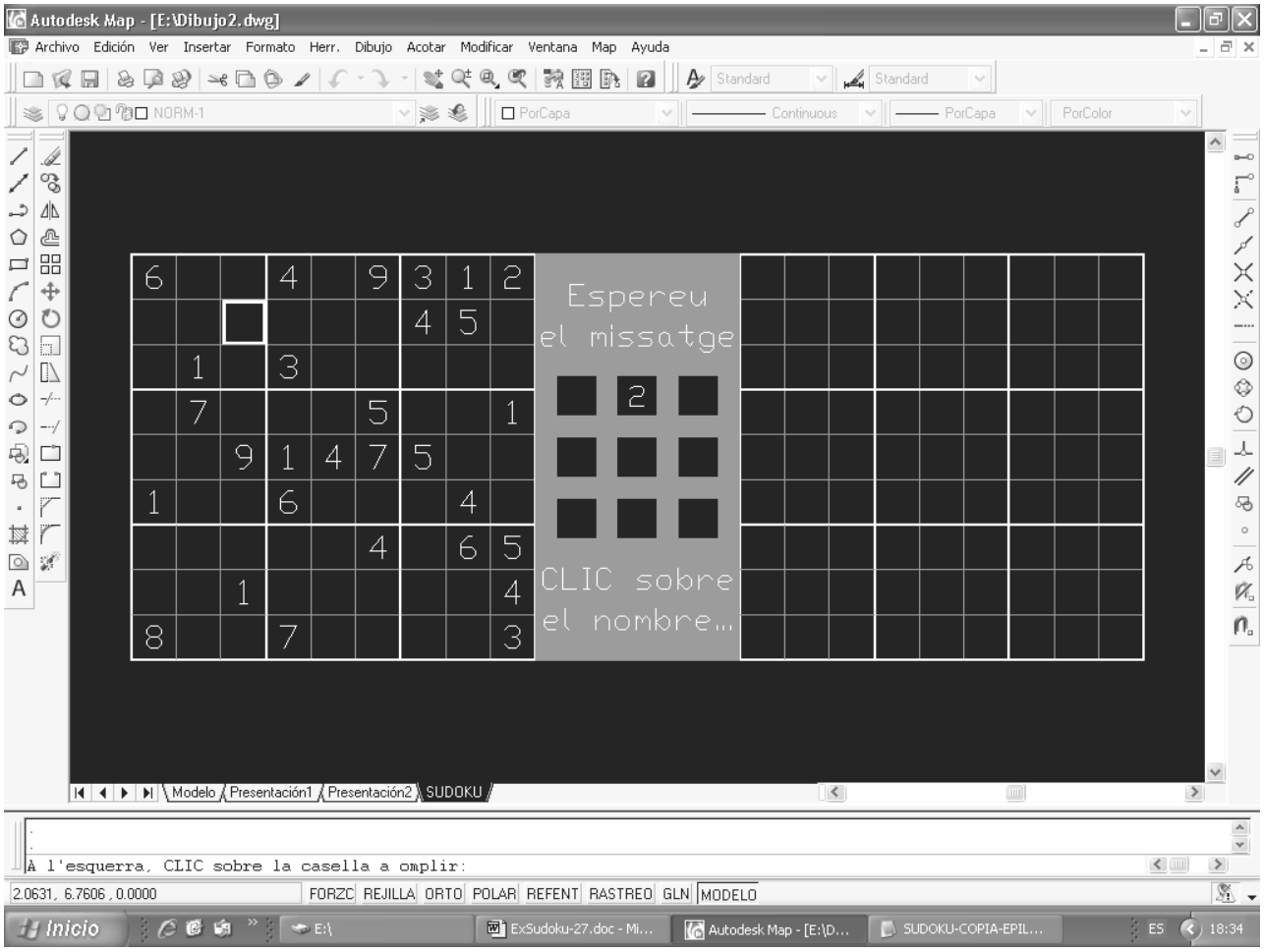
Amb l'opció A, en activar la casella 7,8 es posa en marxa el dispositiu de ...



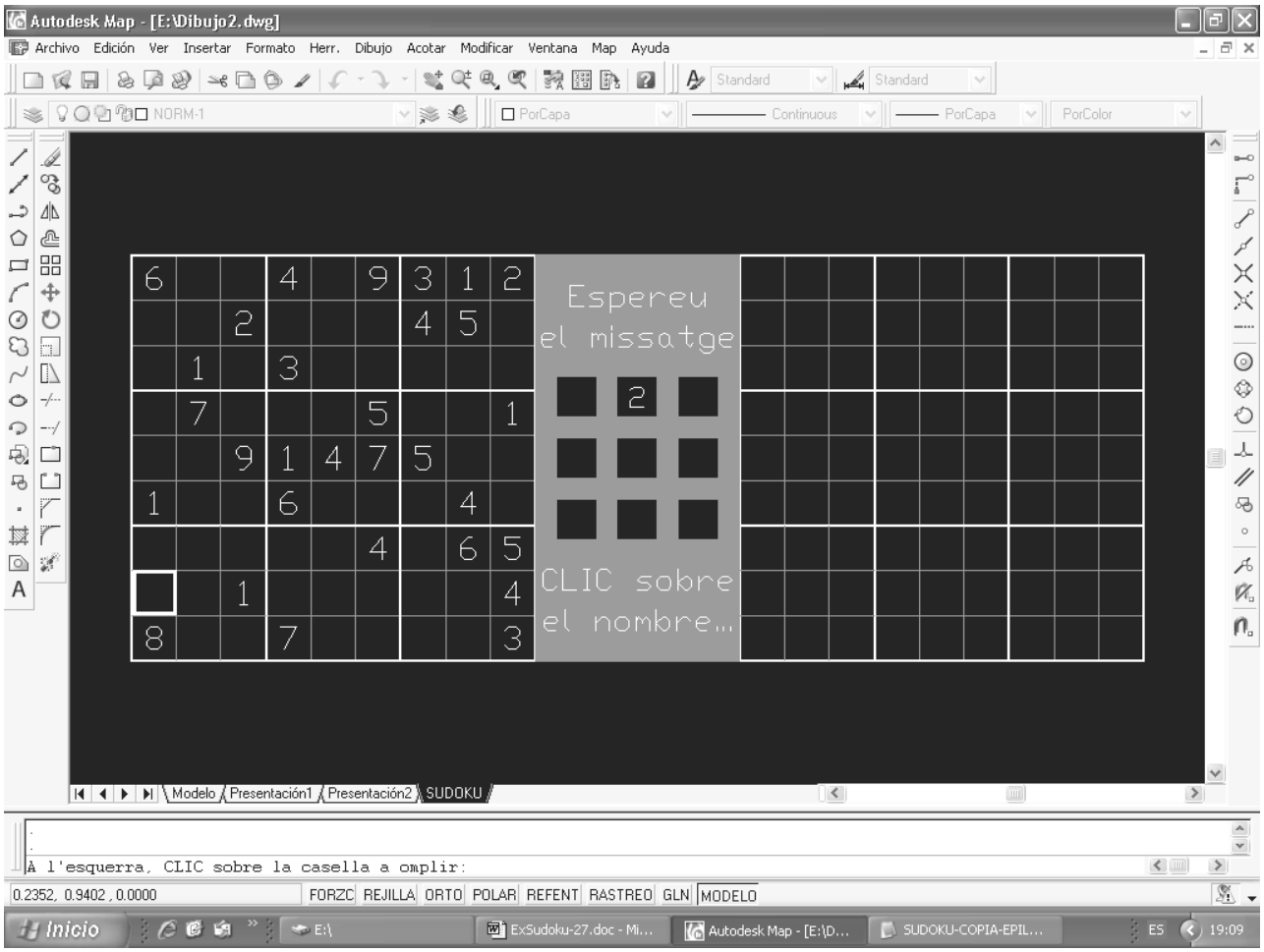
control no trivial de compatibilitat. Després d'emplenar la casella 5,5, els...



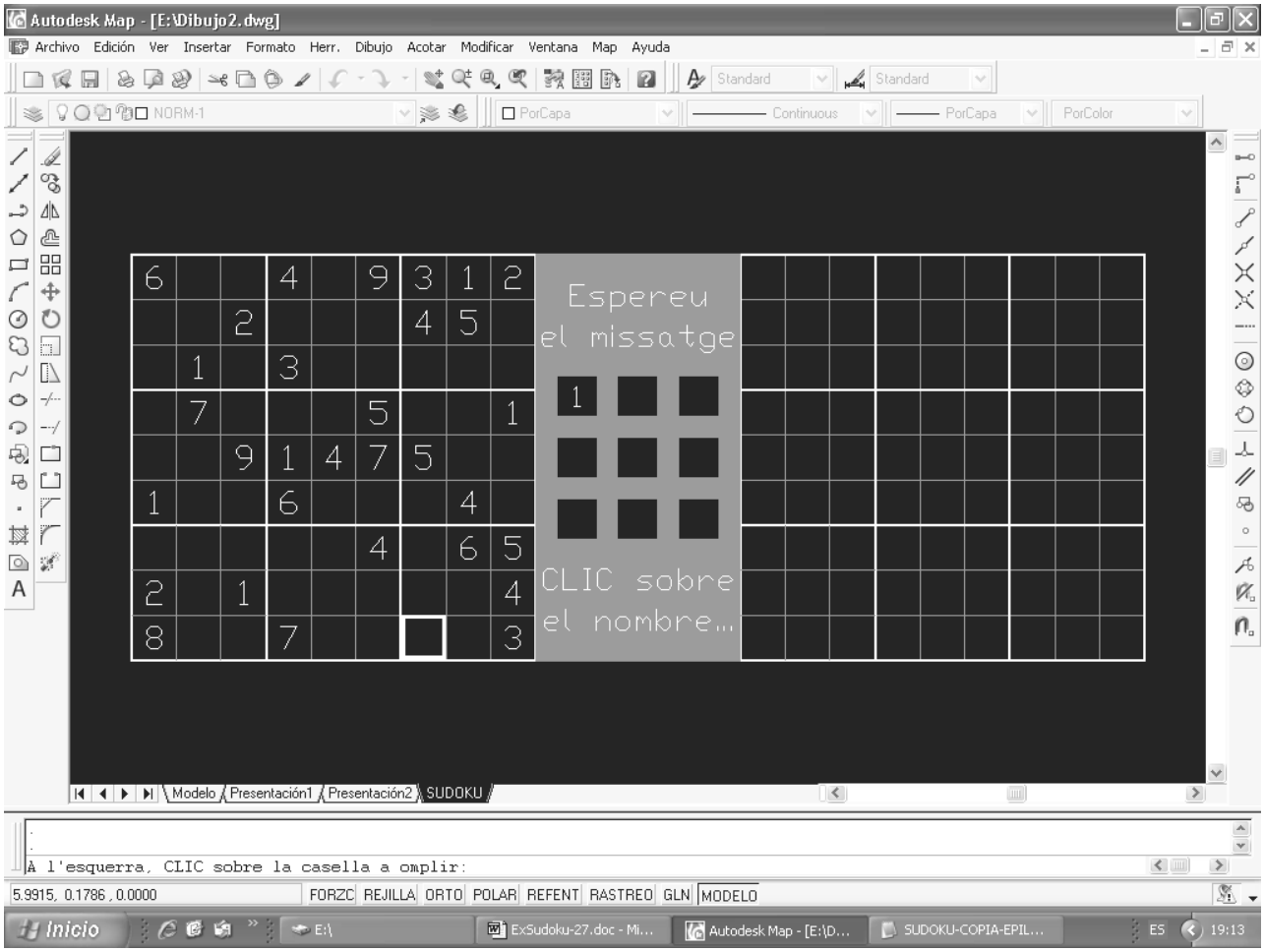
restants emplenaments queden determinats, en tenir un únic candidat, però cal...



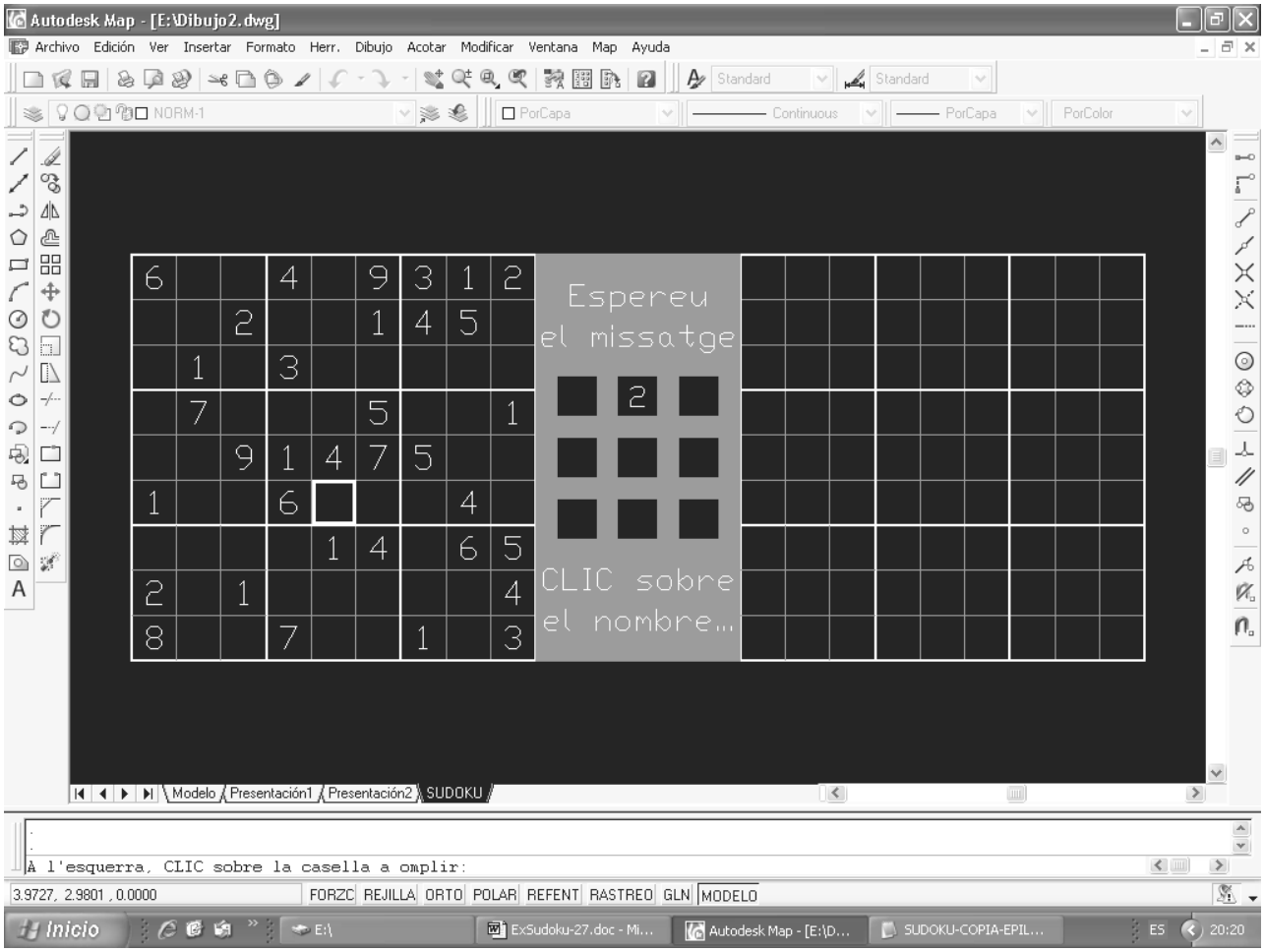
emplenar-ne manualment uns quants més (les caselles 9,6; 3,8; 1,2; 7,1 i 5,4)...



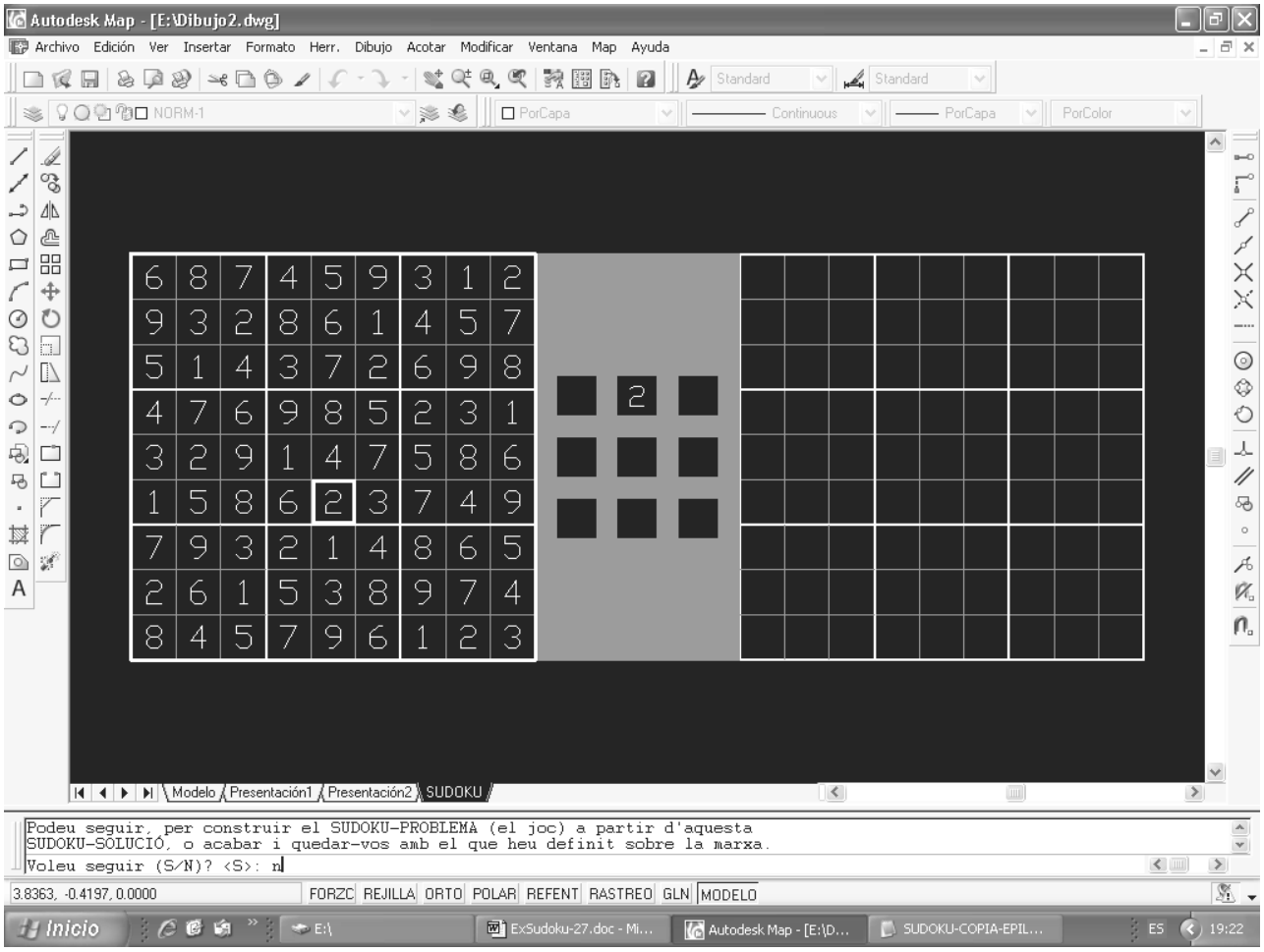
perquè es defermi la reacció en cadena d'emplenaments automàtics, que assolirà...



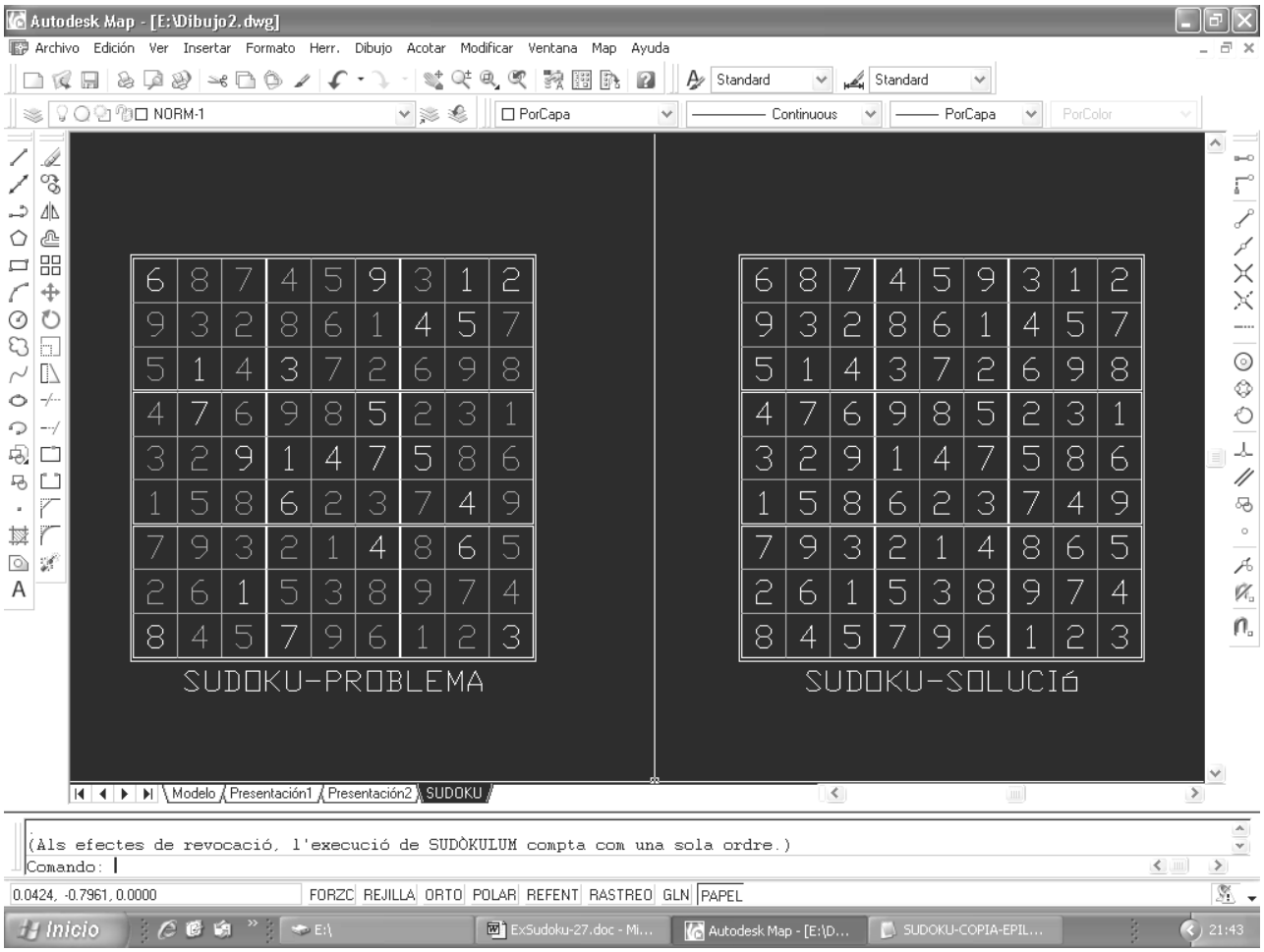
la SUDOKU-SOLUCIÓ. Cal dir que el nombre d'emplenaments manuals suplementaris...



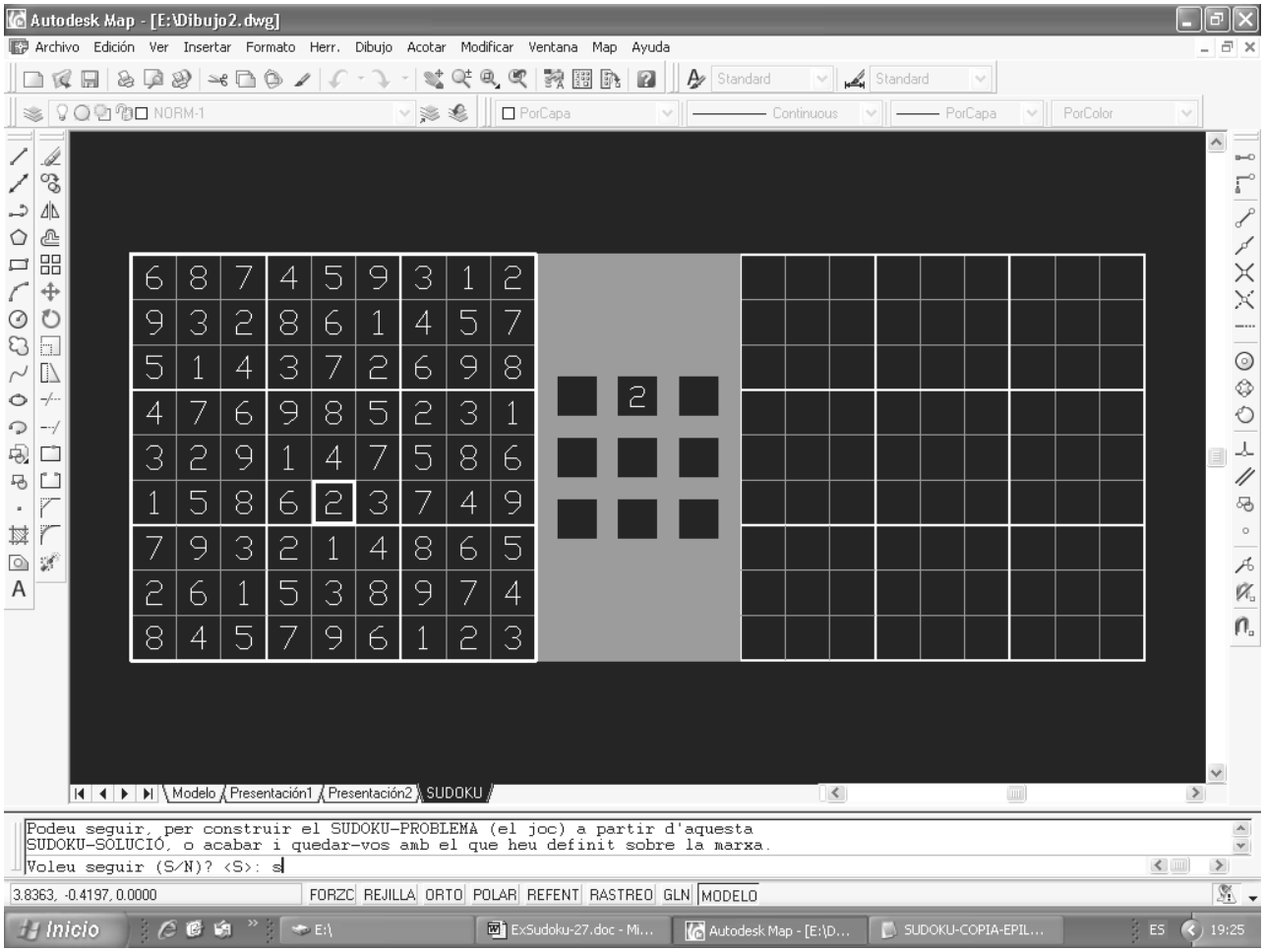
anirà disminuint en nombre en les pròximes versions, fins quedar reduïts a zero...



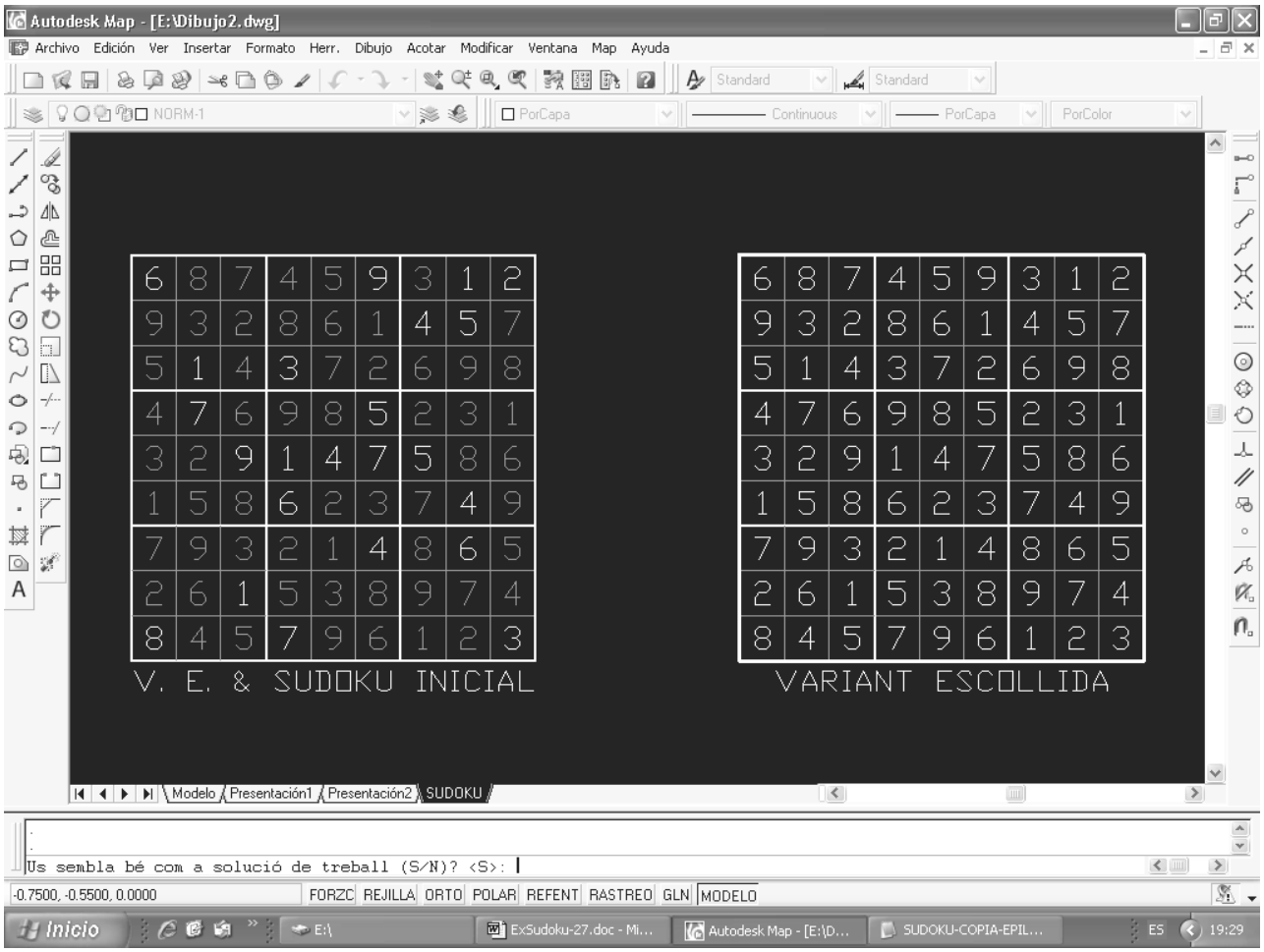
(és el cas d'aquest sudoku) o a molt pocs. Tornant a la consecució de la SUDOKU-



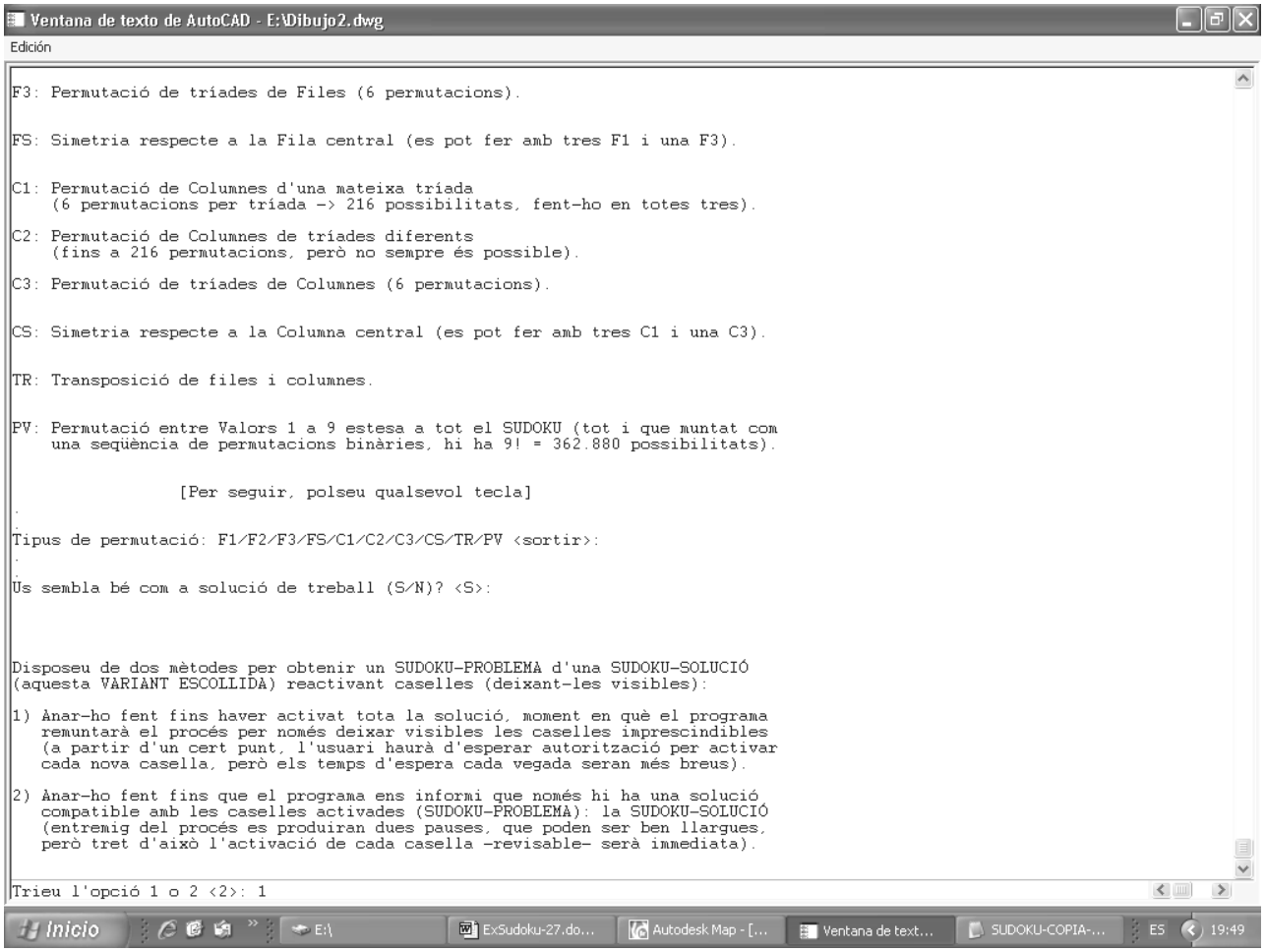
SOLUCIÓ, hi ha la possibilitat de plantar-se, acceptant el SUDOKU-PROBLEMA...



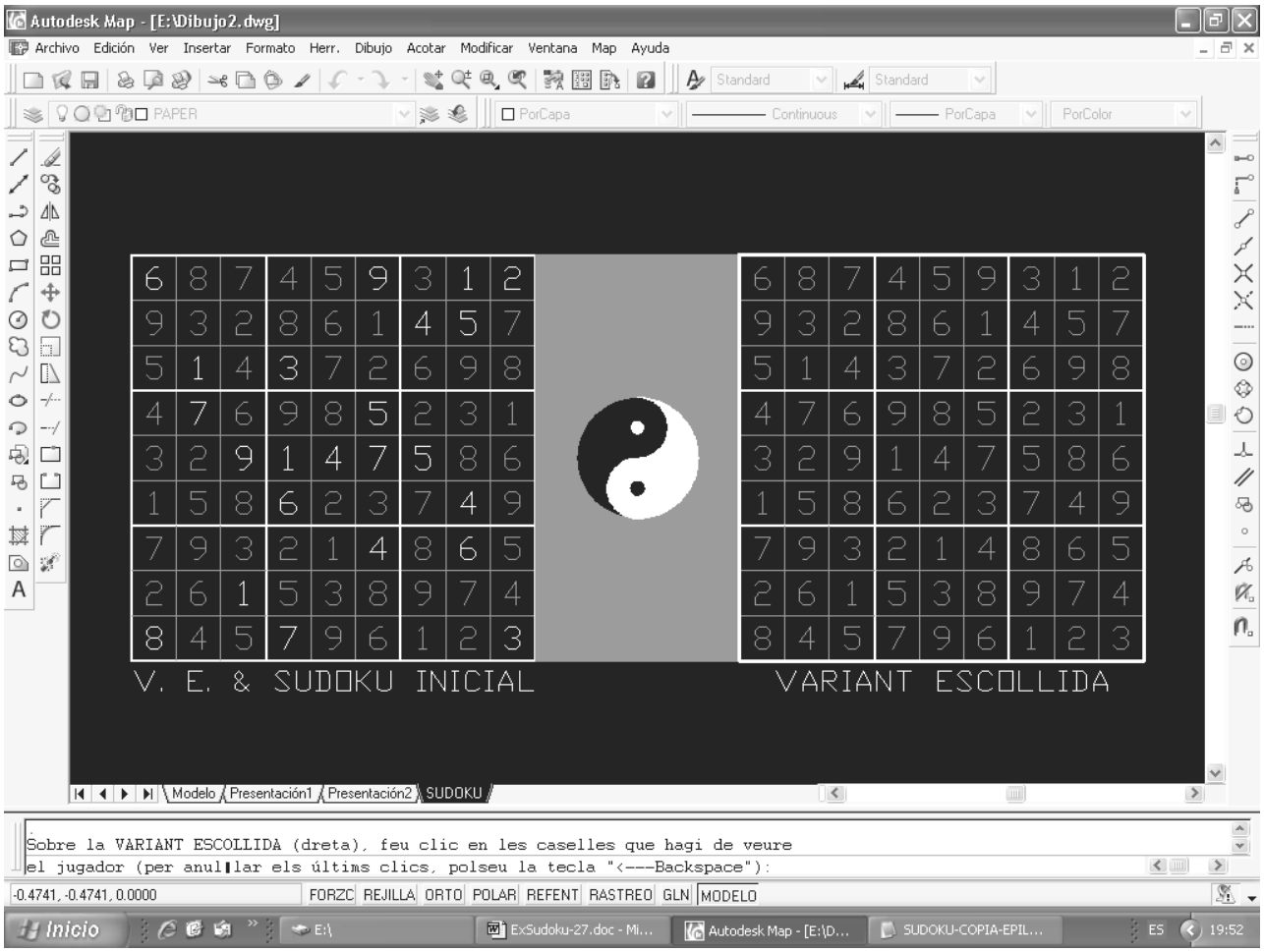
definit per les caselles emplenades manualment i l'ordre seguit, o continuar, ...



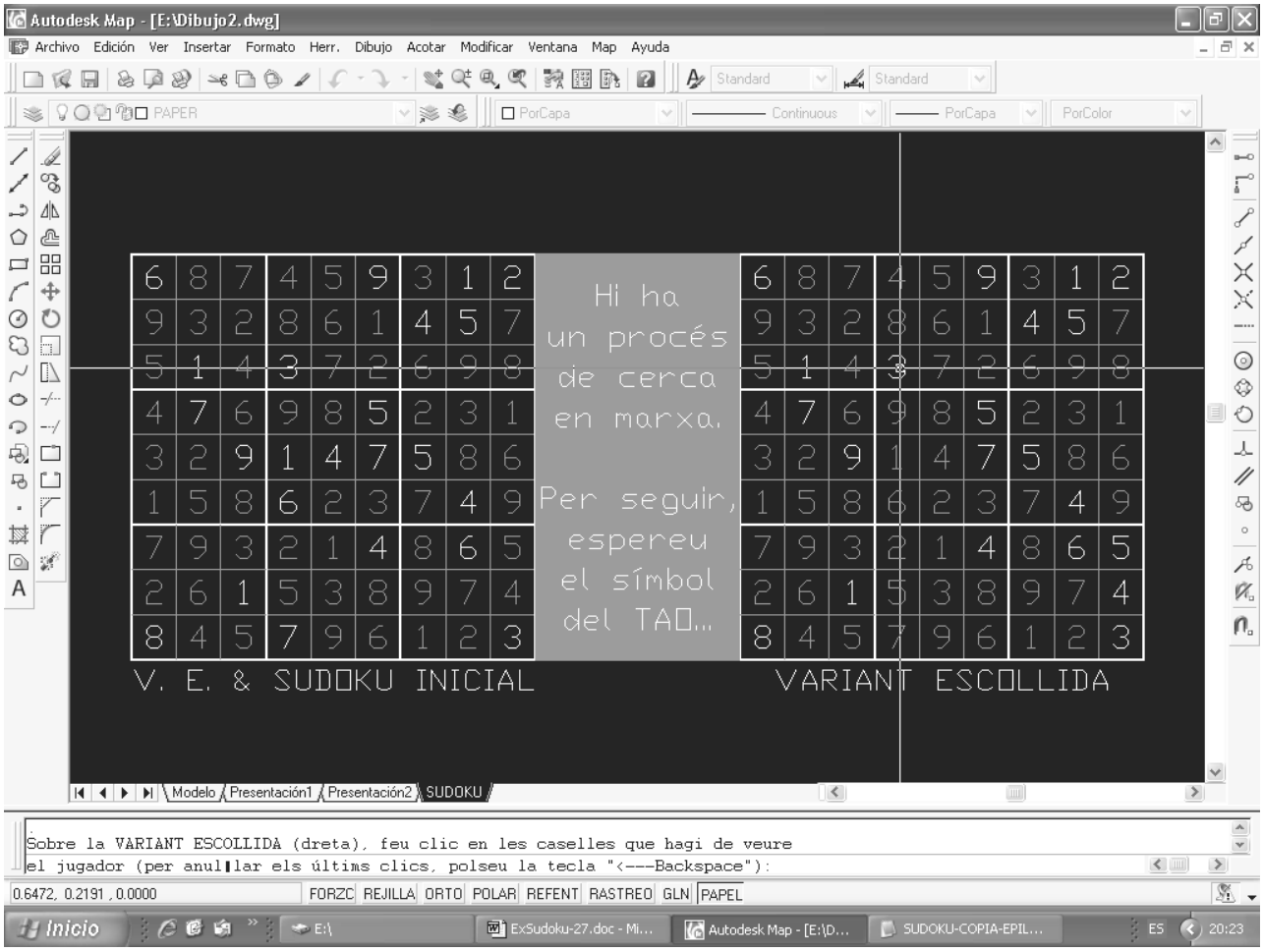
aplicant alguna transformació, si s'escau (aquí no ho fem) i, si donem per bona...



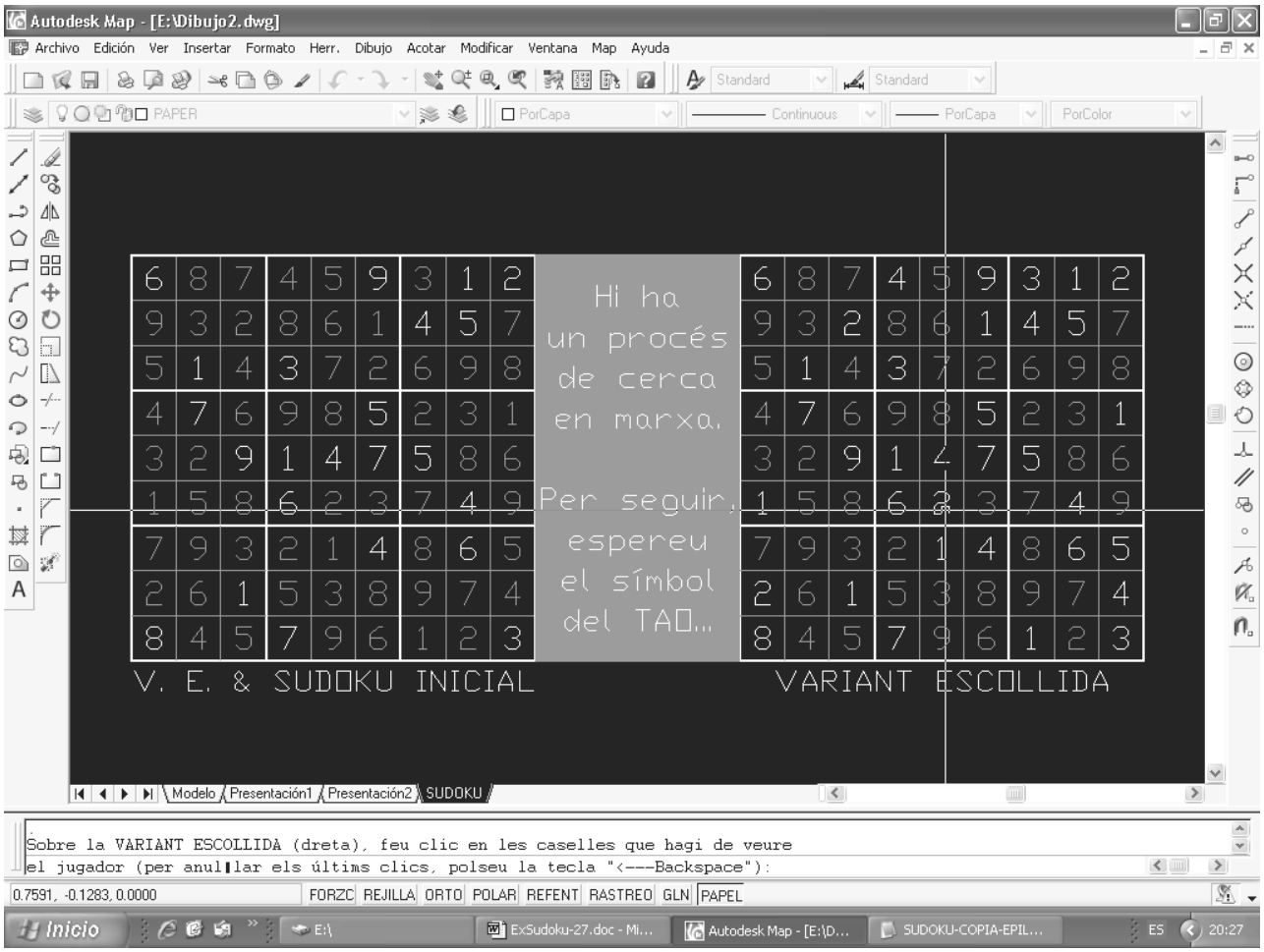
la VARIANT ESCOLLIDA, obtenir un SUDOKU-PROBLEMA amb un dels dos mètodes oferts.



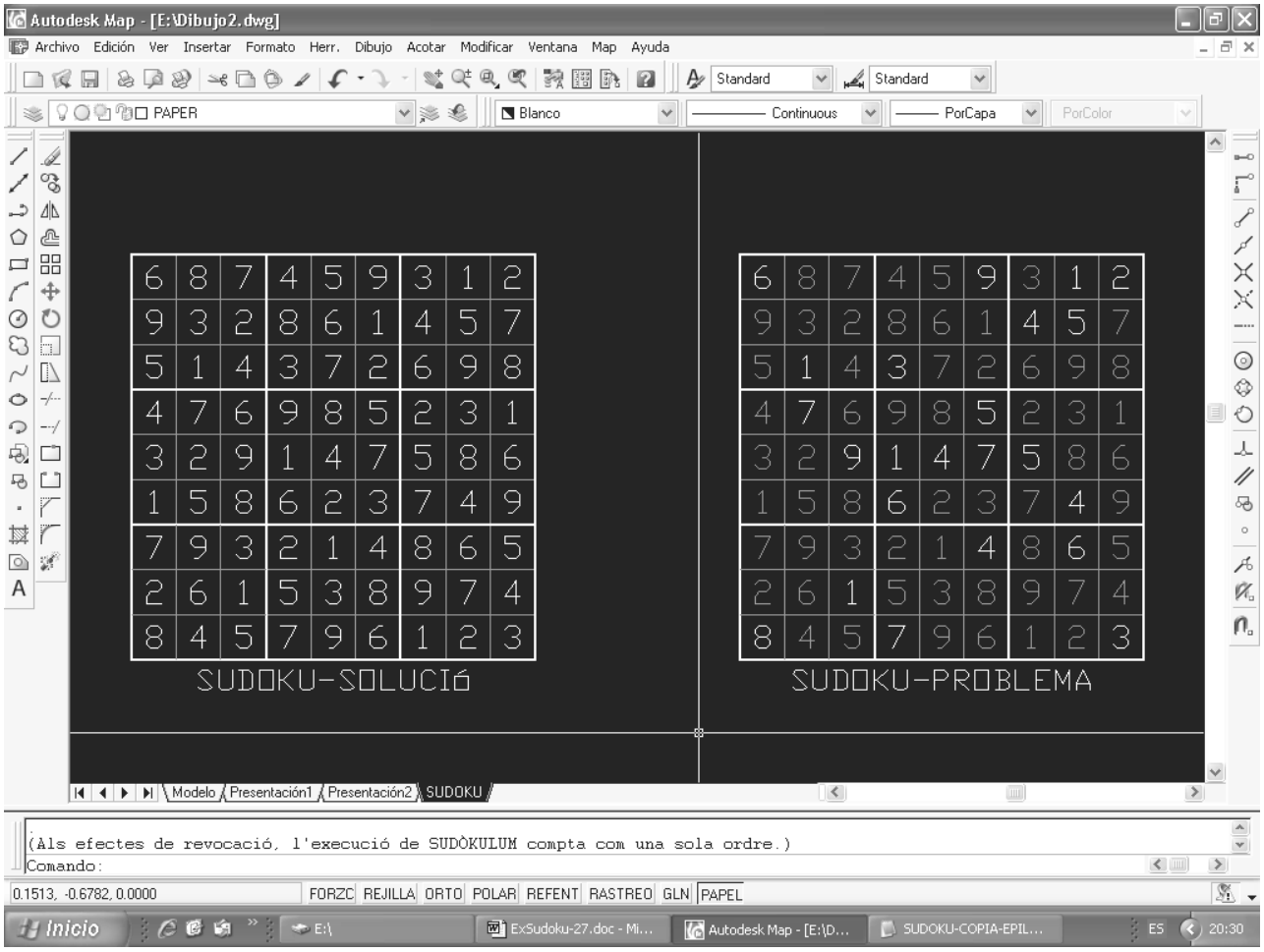
Com que el 2 ja l'hem vist, utilitzarem el mètode 1. A la pràctica es tractaria...



precisament d'obtenir un SUDOKU-PROBLEMA diferent de l'inicial, però a títol de...



comprovació repetirem els emplenaments d'abans. Amb la casella 4,7 s'activa un...



control discret però que, com abans, obliga a marcar 28 caselles per aïllar-ne 23.

L'adopció d'alguns dels canvis formals decidits pàgines enrera, en aquest mateix capítol, ha permès efectuar simplificacions que, de retruc, n'han portat d'altres (i, si l'autor no estigués ja fart de revisions obligades, probablement a cada nou repàs aniria trobant-ne més, però és de bon cristià no escurar massa el plat i cal deixar unes quantes engrunes per als qui s'hi vulguin entretenir). A **FES-SOLUCIO**, per exemple, la voluntat de limitar a la finestra de l'esquerra la representació gràfica dels emplenaments discrecionals (opció A, a la primera meitat), ha permés escurçar considerablement la intervenció de l'editor de dibuixos, que de ser

```
.....................................................................
(TECLA-P ())
(command "ESPACIOM" "CVPORT" 2
         "CAPA" "D" "NORM-1" "" "TEXTO" "MC" P 0.5 0 (itoa N)
         "CVPORT" 3
         "CAPA" "D" "VAR-1" ""  "TEXTO" "MC" P 0.5 0 (itoa N)
         "CVPORT" 2)
(setq PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
.....................................................................
```

ha passat a

```
.....................................................................
(TECLA-P ())
(command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
(setq PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
.....................................................................
```

i també la funció **PLUS-N->P**, que semblava condemnada a tenir aquest gruix,

```
(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (if 1-2
          (progn
            (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
                  PPM (cons (list P1 P2 0) PPM)
                  LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                                LLL/LLL))
            (command "COPIA" "C" P1 P2 "" "0,0" ""
                     "CAMBPROP" "LT" "" "C" "VAR-1" ""))
          (command "ESPACIOM"  "CVPORT" 2
                   "CAPA" "D" "NORM-1" "" "TEXTO" "MC" A-P 0.5 0 (itoa N)
                   "CVPORT" 3
                   "CAPA" "D" "VAR-1" ""  "TEXTO" "MC" A-P 0.5 0 (itoa N)
                   "CVPORT" 2)))))
```

l'hem poguda aprimar considerablement, deixant-la així

```
(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (if 1-2
          (progn
            (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
                  PPM (cons (list P1 P2 0) PPM)
                  LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                                LLL/LLL))
            (command "COPIA" "C" P1 P2 "" "0,0" ""
                     "CAMBPROP" "LT" "" "C" "VAR-1" ""))
          (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N)))))
```

Però a més, aquesta segona mirada a **FES-SOLUCIO** ens ha fet descobrir a l'opció B, on no calia tocar res perquè ja ens estava bé veure l'emplenament des d'ambdues finestres, enduts per la rutina havíem dibuixat els valors per partida doble, a la capa NORM-1 (des de la finestra esquerra) i a la capa VAR-1 (des de la dreta),

```
.....................................................................
(TECLA-P ())
(command "DESPLAZA" CURSOR "" "0.1,0" "" "ESPACIOM"
         "CVPORT" 2
         "CAPA" "D" "NORM-1" ""
         "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
         "CVPORT" 3
         "CAPA" "D" "VAR-1"  ""
         "TEXTO" "MC" (list J K) 0.5 0 (itoa N) "ESPACIOP")
(setq L (append L (list N))))
.....................................................................
```

sense adonar-nos que, una vegada completat l'emplenament, l'accés a **GRAF-9*9** des de **CANON->VARIANT** ja s'encarregaria de posar cada cosa en el seu lloc (encara que optéssim per no aplicar a ****9*9**** cap transformació) i que mentrestant ens podíem limitar a dibuixar-los en una capa visible des d'ambdues finestres, com PAPER (on el fet de servir de suport habitual a objectes gràfics pertanyents l'Espai Paper no impedia de fer-ho ocasionalment sobre l'Espai Model), amb l'economia que veieu:

```
            ..........................................
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.1,0" "" "ESPACIOM"
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N) "ESPACIOP")
            (setq L (append L (list N))))
            ..........................................
```

Fer el mateix amb l'opció C, on l'assignació en bloc a ****9*9**** i ****9**9**** permetia aprofitar l'existència de **GRAF-9*9**, ja no resultava pràctic.

En altres casos les simplificacions (almenys pel que fa a l'extensió del codi) no han tingut res a veure amb el cop de timó efectuat el present capítol, sinó que, com la funció **RETOL-2**, que ja en el capítol *Tornem a començar: de la solució al problema* qualificàvem de totxo insubstancial amb 9 referències **RETOL-1** però que potser anava bé que inicialment fos

```
(defun RETOL-2 (T1 T2 T3 T4 T5 T6 T7 T8 T9)
  (RETOL-1 '(0 0.4)  T1) (RETOL-1 '(0 0.3)  T2) (RETOL-1 '(0 0.2)  T3)
  (RETOL-1 '(0 0.1)  T4) (RETOL-1 '(0 0)    T5) (RETOL-1 '(0 -0.1) T6)
  (RETOL-1 '(0 -0.2) T7) (RETOL-1 '(0 -0.3) T8) (RETOL-1 '(0 -0.4) T9))
```

per facilitar-ne la comprensió, després no ens ha importat desdir-nos d'allò que llavors havíem assegurat (que es representaven amb la seva formulació definitiva) per anar a una versió més compacta:

```
(defun RETOL-2 (T1 T2 T3 T4 T5 T6 T7 T8 T9)
  (setq K 0.5)
  (foreach E (list T1 T2 T3 T4 T5 T6 T7 T8 T9)
    (RETOL-1 (list 0 (setq K (- K 0.1))) E)))
```

Com a última precisió, afegirem que els noms assignats en el capítol 1 (*Tornem a començar: de la solució al problema*) a les capes utilitzades des de l'Espai Model (NORM-VAR, NORM-0, NORM-1, VAR-0 i VAR-1) potser no eren els més adients, vista a hores d'ara la varietat d'usos adoptats per les finestres esquerra i dreta, però els mantindrem malgrat tot. I, ja sense més preàmbuls, us oferim el codi íntegre (dels comentaris inclosos en funcions que havien estat presentades en el capítol esmentat, únicament repetirem els d'aquelles que han sofert alguna modificació):

```
(defun *** () (terpri) (repeat 78 (prompt "*")))

(defun DIB-3*3 (Z)
  (command "SCP" "DE" "1,1"
           "RECTANG" "-1.485,-1.485" "1.485,1.485")
  (if Z (command "ZOOM" "E" "ZOOM" "0.9X"))
  (command "DESCOMP" "LT"
           "MATRIZ" "0,-1.485" "" "R" 3 1 0.99
           "MATRIZ" "-1.485,0" "" "" 1 3 0.99))

(defun TRIA-G ()
  (command "DESHACER" "M"
           "VENTANAS" ""
           "ESPACIOM"
           "ZOOM" "C" "7,4.45" 15.8)
  (repeat 3
    (setq G (1+ G))
    (DIB-3*3 ())
    (command "TEXTO" "MC" "-1,-1" "0.5" "0" "1"
             "TEXTO" "MC" "0,-1" "" "" "2"
             "TEXTO" "MC" "1,-1" "" "" "3"
             "TEXTO" "MC" "-1,0" "" "" "4"
             "TEXTO" "MC" "0,0" "" "" "5"
             "TEXTO" "MC" "1,0" "" "" "6"
             "TEXTO" "MC" "-1,1" "" "" "7"
             "TEXTO" "MC" "0,1" "" "" "8"
             "TEXTO" "MC" "1,1" "" "" "9"
```

```
            "CAMBPROP" "OC" "-1,0" "0,1" "1,0" "0,-1" ""
                       "E" "C" "-0.2,-0.2" "0.2,0.2" "" "O" G ""
            "RECTANG" "1.515,-1.485" "13.485,1.485"
            "COLOR" G "SOMBREA" "S" "LT" "" "COLOR" "PORCAPA"
            "TEXTO" "MC" "7.5,1" "" "" "Cal diferenciar prou"
            "TEXTO" "MC" "7.5,0" "" "" "1-3-5-7-9 de 2-4-6-8"
            "TEXTO" "MC" "7.5,-1" "" "" "i llegir bé aquest text."
            "SCP" "DE" "-1,2.45"))
  (while (not (and (= (car (setq GR (grread))) 3)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (> PX -0.485) (< PX 14.485)
                   (setq G (cond ((and (> PY -10.835) (< PY -7.865)) 252)
                                 ((and (> PY  -7.385) (< PY -4.415)) 253)
                                 ((and (> PY  -3.935) (< PY -0.965)) 254))))))
  (command "DESHACER" "R"
           "TILEMODE" 1))

(defun PREPGRAF-9*9 ()
  (setq G 251                ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        SVT (getvar "SAVETIME")
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
           "SAVETIME" 0
           "SORTENTS" 127
           "FILLMODE" 1
           "OSMODE" 0
           "CAPA" "E" "PAPER" "I" "*"
;                 "C" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "O" 7       ; ACAD 2000
                  "CR" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "CO" 7     ; ACAD 2004
                  "PAPER,NORM-VAR,NORM-1,VAR-1" "D" "NORM-VAR" ""
           "COLOR" "PORCAPA"
           "TIPOLIN" "D" "CONTINUOUS" ""
           "LWDISPLAY" 0
           "ESTILO" "STANDARD" "TXT" 0 1 0 "N" "N" "N"
           "TILEMODE" 1
           "VENTANAS" "N"
           "SCP" "" "PLANTA" ""
           "SIMBSCP" "DES"
           "MODOSOMBRA" "2D"
           "PRESENTACION" "N" "SUDOKU"
           "PRESENTACION" "D" "SUDOKU"
           "SIMBSCP" "DES")
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvingut/Benvinguda al programa SUDÒKULUM!")
  (prompt "  ****************")
  (***) (***)
  (prompt "\n\n\nSUDÒKULUM no ha estat concebut per resoldre sudokus (tot i que ")
  (prompt "pot usar-se amb\naquest propòsit, renunciant al goig de jugar-hi com ")
  (prompt "Déu mana) sinó per fer-ne.\n\nFunciona sobre AutoCAD 2004, i supera ")
  (prompt "altres productes similars en que la seva\nlentitud exasperant obliga")
  (prompt "rà l'usuari a exercitar la virtut de la paciència...\n\nConvé aclarir")
  (prompt " que SUDÒKULUM no té res a veure amb CURRÍCULUM ni amb ESPÈCULUM.\n\n")
  (repeat 68 (prompt " "))
  (prompt "Joan Colom\n\n\nAbans que res, cal fer quatre ajustos: ")
  (prompt "\n\n1) Amplieu al màxim la finestra de text [Per seguir, polseu ")
  (prompt "qualsevol tecla]")
  (grread)
  (graphscr)
  (prompt "\n.\n2) Deixeu tres línies de text inferiors [Per seguir, polseu ")
  (prompt "qualsevol tecla]")
```

```
  (grread)
  (prompt "\n.\n3) Obriu la pàgina \"Visual.\" i en el requadre \"Elementos de ")
  (prompt "presentación\"\n   deixeu activada únicament la 1ª casella ")
  (prompt "[Per seguir, piqueu en \"Aceptar\"]\n")
  (command "OPCIONES")
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: aquella en què el ")
  (prompt "color GRIS es\n   diferencii bé del BLANC i el NEGRE, i doni bon ")
  (prompt "contrast entre text i fons.")
  (TRIA-G)
  (DIB-3*3 T)
  (command "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
           "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
           "CAPA" "B" "NORM-VAR"
;                  "C" G "NORM-0,VAR-0" "D" "PAPER" ""                  ; ACAD 2000
                   "CO" G "NORM-0,VAR-0" "D" "PAPER" ""                 ; ACAD 2004
           "PRESENTACION" "D" "SUDOKU"
           "VENTANAS" "-1.25,-0.5" "-0.25,0.5"
           "VENTANAS" "0.25,-0.5" "1.25,0.5"
           "ZOOM" "E" "ZOOM" "0.9X"
           "ESPACIOM"
;          "CVPORT" 2                                                   ; ACAD 2000
           "CVPORT" 2 "UCSVP" 0                                         ; ACAD 2004
           "VGCAPA" "I" "VAR-1,VAR-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
;          "CVPORT" 3                                                   ; ACAD 2000
           "CVPORT" 3 "UCSVP" 0                                         ; ACAD 2004
           "VGCAPA" "I" "NORM-1,NORM-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
           "SCP" "" "ESPACIOP"
           "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))

(defun PREPTEXT-9*9 ()
  (textscr)
  (prompt "\n\nAquest programa permet crear un SUDOKU-PROBLEMA a partir d'una ")
  (prompt "SUDOKU-SOLUCIÓ,\na base de decidir quines caselles es faran públiques")
  (prompt " (visibles per al jugador).\nPel que fa a la SUDOKU-SOLUCIÓ:")
  (prompt "\n\nA) Podeu improvisar-la, emplenant casella a casella sota control ")
  (prompt "del programa.")
  (prompt "\n\nB) Podeu adoptar-ne una de coneguda, que caldrà transcriure ")
  (prompt "casella a casella.")
  (prompt "\n\nC) Si la solució us és indiferent podeu usar aquesta, que ")
  (prompt "anomenarem CANÒNICA:")
  (prompt "\n\n                                9 1 2  3 4 5  6 7 8")
  (prompt   "\n                                6 7 8  9 1 2  3 4 5")
  (prompt   "\n                                3 4 5  6 7 8  9 1 2")
  (prompt "\n\n                                8 9 1  2 3 4  5 6 7")
  (prompt   "\n                                5 6 7  8 9 1  2 3 4")
  (prompt   "\n                                2 3 4  5 6 7  8 9 1")
  (prompt "\n\n                                7 8 9  1 2 3  4 5 6")
  (prompt   "\n                                4 5 6  7 8 9  1 2 3")
  (prompt   "\n                                1 2 3  4 5 6  7 8 9")
  (initget "A B C")
  (setq ABC (getkword "\n\nTrieu l'opció A, B o C <C>: ") ABC (if ABC ABC "C"))
  (graphscr))

(defun TECLAT ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.2,-0.2" "-0.1,-0.1"
           "COPIA" "LT" "" "-1.0419,0.5919" ""
           "EDITPOL" (setq CURSOR (ssget "_L")) "G" 0.006 ""
           "MATRIZ" "-0.1,-0.1" "" "R" 3 3 0.15 0.15
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "COLOR" G
           "SOMBREA" "S" "C" "-0.22,-0.53" "0.22,0.53" ""
           "COLOR" "PORCAPA")
  (MASCARA L1A9)
```

```
  (if (and (= ABC "A") (not RESP) (not 1-2))
      (command "ESPACIOM" "CVPORT" 2 "CAPA" "D" "NORM-1" "")))

(defun INI-COORDS ()
  (setq Y -1 PY ())
  (reverse (repeat 9
             (setq Y (1+ Y) X -1 PX ()
                   PY (cons (reverse (repeat 9
                                       (setq X (1+ X) PX (cons (list X Y) PX))))
                            PY)))))

(defun INI-LLL ()
  (setq LL () LLL ())
  (repeat 9 (setq LL (cons L1A9 LL)))
  (repeat 9 (setq LLL (cons LL LLL))))

(defun COMPLET-9*9 (9*9 / COMP)
  (setq COMP T)
  (foreach L 9*9
    (if COMP (foreach E L
               (if (and COMP (listp E)) (setq COMP () R-P (if POST E))))))
  COMP)

(defun ELEMENT (LL) (nth X (nth Y LL)))

(defun TECLA-M ()
  (while (not (and (= (car (setq GR (grread))) 3)
                   (= (getvar "CVPORT") 2)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (equal (list PX PY) '(4 4) 4.48)
                   (or (< (- PX (setq X (fix PX))) 0.48)
                       (< (- (setq X (fix (1+ PX))) PX) 0.48))
                   (or (< (- PY (setq Y (fix PY))) 0.48)
                       (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                   (ELEMENT LLL))))
  (setq P (list X Y)))

(defun SUD-MIN (9L PREVI / J0 J1 J2 J3)
  (setq I 0 J 0 K 0 POST T)
  (foreach LL (list 9L (TRANSPOSAR 9L))
    (if (and J1 POST (< I 17)) (setq POST ()))
    (setq J3 1)
    (if POST
        (foreach L LL
          (if POST
              (progn
                (foreach E L
                  (if (or (atom E) (and PREVI (equal E (list X Y))))
                      (progn
                        (if (or (and (not J1) (= J3 1))
                                (and J1 (not J2) (> J3 1)))
                            (setq I (1+ I)))
                        (setq J (1+ J)))))
                (setq J0 (cons J J0))
                (if (< K 2)
                    (setq K (1+ K))
                    (if (or (< J 2)
                            (equal J0 (list J 0 0))
                            (equal J0 (list J J 0))
                            (equal J0 (list J J J)))
                        (setq POST ())
                        (progn
                          (if J1
                              (if J2
                                  (setq J2 (if (< J J2) J J2)
                                        POST (or (< J3 3) (>= (+ J1 J2) 7)))
                                  (if (= J3 1)
```

```
                                        (setq J2 J)
                                        (setq J1 (if (< J J1) J J1))))
                                  (setq J1 J))
                             (setq J0 () J 0 K 0 J3 (1+ J3)))))))))
  (if (and POST (= I 17) (= (min J1 J2) 2)) (setq POST ())))

(defun M->P () (list (+ (* 0.1105 X) -1.1919) (+ (* 0.1105 Y) -0.4419)))

(defun TRANSPOSAR (A / LLE LL LE E LS LLS)
  (setq LLE A)
  (while LLE
         (setq LS () LL LLE)
         (foreach L LL (setq E (car L) LE (cdr L)
                             LS (append LS (list E))
                             LLE (if (not (equal LLE LL)) LLE)
                             LLE (if LE (append LLE (list LE)))))
         (setq LLS (append LLS (list LS)))))

(defun TRANSPOSA-HO ()
  (setq K (if 1-2 **V**V** **9*9**)
        K (TRANSPOSAR K) LLL (TRANSPOSAR LLL) P (reverse P) 9**9 ())
  (foreach EE K
    (setq 0-L ())
    (foreach E EE (setq 0-L (cons (if (atom E) E (reverse E)) 0-L)))
    (setq 9**9 (cons (reverse 0-L) 9**9)))
  (setq 9**9 (reverse 9**9) K X X Y Y K)
  (if 1-2 (setq **V**V** 9**9) (setq **9*9** 9**9)))

(defun F=3 (9*9)
  (setq 9**9 ())
  (foreach J JJ
    (setq K (1- (* 3 J)))
    (repeat 3 (setq K (1+ K) 9**9 (cons (nth K 9*9) 9**9))))
  (reverse 9**9))

(defun RETOL-1 (P R / C)
  (setq C (getvar "CLAYER"))
  (if (wcmatch R "V`.*")
      (command "BORRA" "C" "-0.85,-0.58" "-0.65,-0.52" ""))
  (command "CAPA" "D" "PAPER" "" "TEXTO" "MC" P 0.05 0 R "CAPA" "D" C ""))

(defun RETOL-2 (T1 T2 T3 T4 T5 T6 T7 T8 T9)
  (setq K 0.5)
  (foreach E (list T1 T2 T3 T4 T5 T6 T7 T8 T9)
    (RETOL-1 (list 0 (setq K (- K 0.1))) E)))

(defun 3-1 (I) (if (= I 0) '(1 2) (if (= I 1) '(0 2) '(0 1))))

(defun 3-2 (JK) (if (member 0 JK) (if (member 1 JK) 2 1) 0))

(defun ABC/D (/ F1 F2 F3 F/3 I J1 J2 J3 K1 K2)
  (foreach K1 '(0 1 2)
    (setq F1 (nth K1 3*F))
    (foreach J1 '(0 1 2)
      (setq F/3 (nth J1 F1))
      (if (and (not POST) (= (length F/3) 3))
          (foreach K2 (3-1 K1)
            (setq F2 (nth K2 3*F)
                  F3 (nth (3-2 (list K1 K2)) 3*F))
            (foreach J2 (3-1 J1)
              (setq J3 (3-2 (list J1 J2)))
              (foreach I (nth J2 F2)
                (if (and (not (member I F/3))
                         (not (member I (nth J3 F1)))
                         (not (member I (nth J1 F3)))
                         (< (length (nth J3 F3)) 3))
                    (setq POST T)))))))))
```

```
(defun AB/AB (/ F1 F2 F/3 F12 FF1 FF2 I J1 J2 J3 K1 K2)
  (foreach E COMB32
    (if (not POST)
        (setq K1 (car E) K2 (cadr E)
              F1 (nth K1 3*F) F2 (nth K2 3*F)))
    (foreach F COMB32
      (if (not POST)
          (setq J1 (car F) J2 (cadr F) F12 ()
                FF1 (append (nth J1 F1) (nth J2 F1))
                FF2 (append (nth J1 F2) (nth J2 F2))))
      (if (and (not POST) (> (length FF1) 1) (> (length FF2) 1))
          (progn
            (foreach G FF1 (if (member G FF2) (setq F12 (cons G F12))))
            (if (or (= (setq I (length F12)) 2) (= I 3))
                (progn
                  (setq J (3-2 F) F/3 (nth J (nth (3-2 E) 3*F))
                        J3 (if (= (length F/3) 1) (car F/3)))
                  (if (or (and (= I 2) J3 (not (member J3 F12)))
                          (and (= I 3) (or (not F/3) J3 (member J3 F12))))
                      (setq POST T))))
            (if (and (not POST) (> (length F12) 1))
                (progn
                  (setq X (1- (* 3 J))
                        Y (if (= Y0 -1) '(2 5) (if (= Y0 2) '(-1 5) '(-1 2))))
                  (repeat 3
                    (if (not POST)
                        (progn
                          (setq X (1+ X) I 0 L ())
                          (foreach G Y
                            (repeat 3
                              (setq G (1+ G) L (cons (nth X (nth G 9*9)) L))))
                          (foreach G F12
                            (if (member G L) (setq I (1+ I))))
                          (if (> I 1)
                              (progn
                                (setq I 0 F/3 (nth (3-2 F) (nth (3-2 E) 3*F)))
                                (foreach G F12
                                  (if (member G F/3) (setq I (1+ I))))
                                (if (< I 2) (setq POST T))))))))))))))

(defun DETECTA (ORIG / 9*9 X Y X0 Y0 C0 C1 C2 L0 L1 L2 L12 V W V1 V2 3*F F)
  (setq 9*9 (if ORIG A-9*9 (TRANSPOSAR A-9*9)))
  (foreach Y0 '(-1 2 5)
    (setq Y Y0 3*F ())
    (if (not POST)
        (progn
          (repeat 3
            (setq Y (1+ Y) L0 (nth Y 9*9) F ())
            (foreach X0 '(-1 2 5)
              (setq X X0 J 0 V ())
              (if (not POST)
                  (progn
                    (repeat 3
                      (setq X (1+ X) K (nth X L0))
                      (if (listp K) (setq J (1+ J) C0 X) (setq V (cons K V))))
                    (setq F (cons V F))
                    (if (= J 1)
                        (progn
                          (setq J -1 L ())
                          (repeat 9
                            (setq J (1+ J))
                            (if (or (<= J Y0) (> J (+ 3 Y0)))
                                (setq K (nth C0 (nth J 9*9))
                                      L (if (atom K) (cons K L) L))))
                          (setq K (3-1 (- Y Y0 1))
                                L1 (nth (+ Y0 1 (car K)) 9*9)
                                L2 (nth (+ Y0 1 (cadr K)) 9*9)
                                X (+ X0 4) J 0 V1 () V2 ())
```

```
                              (repeat 3
                                (setq X (1- X) V1 (cons (nth X L1) V1)
                                                  V2 (cons (nth X L2) V2)))
                              (if ORIG
                                  (progn
                                    (setq J 0 L12 () W ())
                                    (foreach E V1
                                      (if (listp E)
                                          (setq J (1+ J) C1 (car E))
                                          (setq W (cons E W))))
                                    (if (or (and (= J 1) (= C1 C0)) (= J 3))
                                        (progn
                                          (if (= J 3)
                                              (setq L12 L1)
                                              (setq V (append V W)))
                                          (setq J 0 W ())
                                          (foreach E V2
                                            (if (listp E)
                                                (setq J (1+ J) C2 (car E))
                                                (setq W (cons E W))))
                                          (if (or (and (not L12) (= J 3))
                                                  (and L12 (= J 1) (= C2 C0)))
                                              (progn
                                                (if (not L12)
                                                    (setq L12 L2)
                                                    (setq V (append V W)))
                                                (setq W ())
                                                (foreach E L12
                                                  (if (atom E) (setq W (cons E W))))
                                                (foreach E (append W L)
                                                  (if (not (or POST (member E V)))
                                                      (setq POST T)))))))))
                              (if (not POST)
                                  (foreach E L
                                    (if (and (not (or POST (member E L0)
                                                       (member E V1) (member E V2)))
                                             (or (and (member E L1)
                                                      (not (member E L2)))
                                                 (and (member E L2)
                                                      (not (member E L1)))))
                                        (setq POST T)))))))
                (setq 3*F (cons (reverse F) 3*F))) ; Com que la disposició de files
              (if (not POST) (ABC/D))              ; de 3*F és indiferent, no cal fer
              (if (not POST) (AB/AB))))))          ; (setq 3*F (reverse 3*F))

(defun ACT-LLL (X Y A-LLL / X/3 Y/3 LL L J I)
  (setq X/3 (/ X 3) Y/3 (/ Y 3) J -1)
  (foreach F A-LLL
    (setq L () I -1 J (1+ J))
    (foreach E F
      (setq I (1+ I) L (cons (if (= J Y)
                                 (if (= I X) () (subst () A-N E))
                                 (if (= I X)
                                     (subst () A-N E)
                                     (if (and (= (/ I 3) X/3) (= (/ J 3) Y/3))
                                         (subst () A-N E) E)))
                             L)))
    (setq LL (cons (reverse L) LL)))
  (reverse LL))

(defun T+D<12 ()
  (foreach LL (list A-9*9 (TRANSPOSAR A-9*9))
    (if (not POST)
        (foreach Y '(-1 2 5)
          (if (not POST)
              (progn
                (setq J 0 K ())
```

```
                (repeat 3
                  (setq Y (1+ Y))
                  (foreach X (nth Y LL)
                    (if (atom X)
                        (setq J (1+ J) K (if (member X K) K (cons X K))))))
                (if (>= (+ J (length K)) 12) (setq POST T)))))))
  (if (not POST) (DETECTA T))
  (if (not POST) (DETECTA ())))

(defun SUB-X*Y (X*Y P1 P2 / J K L LL)
  (setq K -1)
  (foreach Y X*Y
    (setq K (1+ K) J -1)
    (if (not (or (< K (cadr P1)) (> K (cadr P2))))
        (progn
          (foreach X Y
            (setq J (1+ J))
            (if (not (or (< J (car P1)) (> J (car P2))))
                (setq L (cons X L))))
          (setq LL (cons (reverse L) LL) L ()))))
  (reverse LL))

(defun E-MEMBER (N X*Y P / K E L)
  (setq K -1)
  (foreach Y X*Y
    (if (setq K (1+ K) E (member N Y))
        (setq L (cons (list (+ (car P) (- (length Y) (length E))) (+ (cadr P) K))
                      L))))
  (reverse L))

(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (if 1-2
          (progn
            (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
                  PPM (cons (list P1 P2 0) PPM)
                  LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                                LLL/LLL))
            (command "COPIA" "C" P1 P2 "" "0,0" ""
                     "CAMBPROP" "LT" "" "C" "VAR-1" ""))
          (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N)))))

(defun NC<2 ()
  (if (= NL 2) (setq X2 (1- X1)))
  (setq X1 (if (= NC 1) (caar CC)) I 0)
  (foreach Y (if (= NL 1) 2Y (list Y1))
    (if (= NL 1) (setq X2 (1- X)))
    (repeat 3
      (if (and (/= (setq X2 (1+ X2)) X1)
               (listp (setq J (nth X2 (nth Y A-9*9)))))
          (setq K J I (1+ I)))))
  (if (= I 1) (PLUS-N->P)))

(defun ACTUALITZA-PLUS (ORIG / N LL CC NL NC 2X 2Y X1 X2 Y1 Y2)
  (foreach N L1A9
    (foreach Y1 '(0 3 6)
      (if (not PLUS)
          (progn
            (setq LL (E-MEMBER N (SUB-X*Y A-9*9 (list 0 Y1) (list 8 (+ Y1 2)))
                                                            (list 0 Y1))
                  NL (length LL))
            (if (= NL 1)
                (progn
                  (setq 2X (mapcar '* (3-1 (/ (caar LL) 3)) '(3 3))
                        2Y (mapcar '+ (3-1 (rem (cadar LL) 3)) (list Y1 Y1)))
```

```
                  (foreach X 2X
                    (if (and (not PLUS)
                             (setq CC (E-MEMBER N (SUB-X*Y A-9*9 (list X 0)
                                                           (list (+ X 2) 8)) (list X 0))
                                   NC (length CC))
                             (< NC 2)
                             (or ORIG (= NC 0)))
                        (NC<2))))
                (if (= NL 2)
                    (progn
                      (setq X1 (* (3-2 (list (/ (caar LL) 3)
                                             (/ (caadr LL) 3))) 3)
                            Y1 (+ Y1 (3-2 (list (rem (cadar LL) 3)
                                                (rem (cadadr LL) 3))))
                            CC (E-MEMBER N (SUB-X*Y A-9*9 (list X1 0)
                                                   (list (+ X1 2) 8)) (list X1 0))
                            NC (length CC))
                      (if (< NC 3)
                          (if (= NC 2)
                              (if ORIG
                                  (progn
                                    (setq X1
                                          (+ X1 (3-2 (list (rem (caar CC) 3)
                                                           (rem (caadr CC) 3)))))
                                    (if (listp (setq K (nth X1 (nth Y1 A-9*9))))
                                        (PLUS-N->P))))
                              (NC<2)))))))))
  (if (and ORIG (not PLUS))
      (foreach Y1 '(0 3 6)
        (if (not PLUS)
            (foreach X1 '(0 3 6)
              (if (not PLUS)
                  (progn
                    (setq CC (SUB-X*Y A-9*9 (list X1 Y1) (list (+ X1 2) (+ Y1 2)))
                          I 0 LL ())
                    (foreach C CC
                      (foreach J C
                        (if (listp J)
                            (setq I (1+ I) K J)
                            (setq LL (cons J LL)))))
                    (if (= I 1)
                        (foreach N L1A9
                          (if (not (or PLUS (member N LL))) (PLUS-N->P)))))))))))
  (if (not PLUS)
      (foreach LL A-9*9
        (if (not PLUS)
            (progn
              (setq I 0)
              (foreach J LL
                (if (listp J) (setq I (1+ I) K J)))
              (if (= I 1)
                  (foreach N L1A9
                    (if (not (member N LL)) (PLUS-N->P)))))))))

(defun ACT-9*9 (N P L1 / L2)
  (reverse (foreach F L1 (setq L2 (cons (if (member P F) (subst N P F) F) L2)))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L LL NOU PLUS PPLUS E F)
  (while (not PLUS)
         (setq X (car A-P) Y (cadr A-P)
               A-9*9 (ACT-9*9 A-N A-P A-9*9)
               A-LLL (ACT-LLL X Y A-LLL))
         (if (not (or POST PPLUS))
             (setq NOPLUS (ACT-9*9 A-N A-P NOPLUS)))
         (if (not (or POST 1-2))
             (progn (T+D<12) (setq NOU POST)))
```

```
          (if REAL
              (progn
                (ACTUALITZA-PLUS T)
                (if (not PLUS)
                    (progn
                      (setq A-9*9 (TRANSPOSAR A-9*9))
                      (ACTUALITZA-PLUS ())
                      (setq A-9*9 (TRANSPOSAR A-9*9))))
                (if (and PLUS NOU) (setq POST () NOU ()))))
          (setq PLUS (not PLUS) PPLUS T))
  (list A-9*9 A-LLL))

(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
  (if (not OK)
      (if (COMPLET-9*9 R-9*9)
          (setq OK T)
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (RE-MEMBER (car 2R) (cadr 2R))))))))
  OK)

(defun TREU-RETOL () (repeat 4 (command "BORRA" "LT" "")))

(defun MASCARA (L / INI OK PROU JJ 9**9 0-**9*9** 0-LLL 0-L)
  (if (not 1-2) (setq LL () M 0 SS (ssadd)))
  (if POST
      (progn
        (if C->F (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL)
            (progn
              (setq JJ (if (< Y 6) '(1 0 2) '(2 1 0))
                    0-LLL (F=3 (if 1-2 **V**V** **9*9**)) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**)
                    0-LLL (F=3 LLL) P (list X (rem Y 3)))))))
  (if (and POST 1-2)
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")))
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
    (if (and (member N L)
             (not (or PROU (= N GR)))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if 1-2
              (setq PROU T)
              (progn
                (if POST (TREU-RETOL))
```

```
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
                (command "DESIGNA" (ssadd (entlast) SS) "")))))
        (if (and POST (not 1-2)) (TREU-RETOL))))
  (if POST
      (progn (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
             (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
             (setq P (list X Y)))))

(defun TECLA-P (VARS / FI)
  (while (not (and (or (= (car (setq GR (grread))) 3)
                       (setq FI (and VARS (> (strlen MSG) 51)
                                     (or (equal GR '(2 13)) (equal GR '(2 32))))))
                   (or FI (equal (setq GR (cadr GR)) GR (list (car GR) (cadr GR)))
                                 '(0 0) 0.2))
                   (or FI (setq N (cond ((equal GR '(-0.15  0.15) 0.05) 1)
                                        ((equal GR '( 0     0.15) 0.05) 2)
                                        ((equal GR '( 0.15  0.15) 0.05) 3)
                                        ((equal GR '(-0.15  0   ) 0.05) 4)
                                        ((equal GR '( 0     0   ) 0.05) 5)
                                        ((equal GR '( 0.15  0   ) 0.05) 6)
                                        ((equal GR '(-0.15 -0.15) 0.05) 7)
                                        ((equal GR '( 0    -0.15) 0.05) 8)
                                        ((equal GR '( 0.15 -0.15) 0.05) 9))))
                   (or VARS (= ABC "B") (member N LL))))))

(defun RETOLS ()
  (RETOL-1 (list (if 1-2 0.75 -0.75) -0.55) "SUDOKU-PROBLEMA")
  (RETOL-1 (list (if 1-2 -0.75 0.75) -0.55) "SUDOKU-SOLUCIÓ"))

(defun RASTRE ()
  (if 1-2
      (progn
        (if (= ABC "A")
            (command "CVPORT" 2 "BORRA" "C" "0,0" "8,8" "" "CVPORT" 3))
        (command "CAPA" "DES" "VAR-1" ""
                 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-1" ""
                 "CAPA" "ACT" "VAR-1" ""))
      (setq ABC "D"))
  (foreach E PPM (ssadd (ssname (ssget "_C" (car E) (cadr E)) 0) SS))
  (command "CAMBPROP" "C" "0,0" "8,8" "E" SS "" "C" (if 1-2 "VAR-0" "NORM-0") "")
  (if 1-2
      (progn
        (if (/= ABC "A")
            (command "CVPORT" 2 "CAPA" "DES" "NORM-1" ""
                     "BORRA" "C" "0,0" "8,8" ""
                     "CAPA" "ACT" "NORM-1" ""))
        (command "ESPACIOP" "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" ""))
      (command "ESPACIOP"))
  (RETOLS))

(defun FES-SOLUCIO (/ COMB32 CURSOR P-CURS C->F)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq COMB32 (list '(0 1) '(0 2) '(1 2))
              PPM () **9*9** (INI-COORDS) NOPLUS **9*9**
              P-CURS '(-1.1919 0.4419))
        (INI-LLL)
```

```
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (if (not POST) (SUD-MIN NOPLUS T))
               (command "ESPACIOP" "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (TECLA-P ())
               (command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
               (setq PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
                     LLL (ACTUALITZA **9*9** N P LLL T)
                     **9*9** (car LLL) LLL (cadr LLL)))
        (while (= (last (car PPM)) 1) (setq PPM (cdr PPM)))
        (setq POST () SS (ssadd))
        (prompt "\n.\nPodeu seguir, per construir el SUDOKU-PROBLEMA (el joc) ")
        (prompt "a partir d'aquesta\nSUDOKU-SOLUCIÓ, o acabar i quedar-vos amb ")
        (prompt "el que heu definit sobre la marxa.")
        (initget "Si No")
        (setq CONF (getkword "\nVoleu seguir (S/N)? <S>: ")
              CONF (if CONF CONF "Si"))
        (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
                 "CAMBPROP" "P" "" "C" "VAR-1" "")
        (if (= CONF "Si") (command "ESPACIOP") (RASTRE)))
      (progn
        (prompt "\n.\n.\nCLIC sobre el nombre que va a la casella marcada: ")
        (setq K 9)
        (repeat 9
          (setq L () J -1 K (1- K))
          (repeat 9
            (setq J (1+ J))
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.110465,0" "" "ESPACIOM"
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N) "ESPACIOP")
            (setq L (append L (list N))))
          (command "DESPLAZA" CURSOR "" "-0.9942,-0.1105" "")
          (setq **9*9** (cons L **9*9**)))))
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))

(defun GRAF-9*9 (VAR / I)
  (setq J -1)
  (foreach 9*9 (list **9*9** **9**9**)
    (if (< J 0)
        (command "CVPORT" 3 "BORRA" "C" "0,0" "8,8" ""
                 "CVPORT" 2 "BORRA" "C" "0,0" "8,8" ""
                 "CAPA" "D" (if VAR "NORM-0" "NORM-1") "")
        (progn
          (setq VAR () J -1)
          (command "CVPORT" 3 "CAPA" "D" "VAR-1" "")))
    (while (< (setq J (1+ J)) 9)
           (setq K -1)
           (while (< (setq K (1+ K)) 9)
                  (setq I (nth J (nth K 9*9)))
                  (command "TEXTO" "MC" (list J K) 0.5 0 (itoa (abs I)))
                  (if (and VAR (< I 0))
                      (command "CAMBPROP" "LT" "" "C" "NORM-1" ""))))))

(defun SEGUIR (GRAF)
  (if GRAF (prompt "\n.\n."))
  (prompt "\n                  [Per seguir, polseu qualsevol tecla]")
  (grread))

(defun VARIACIONS ()
  (textscr)
  (repeat 16 (terpri))
  (prompt "D'aquesta SUDOKU-SOLUCIÓ se'n poden treure d'altres, per aplicació ")
  (prompt "reiterada\nde les transformacions següents:")
```

```
  (prompt "\n\n\nF1: Permutació de Files d'una mateixa tríada\n    (6 ")
  (prompt "permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).")
  (prompt   "\n\nF2: Permutació de Files de tríades diferents")
  (prompt "\n    (fins a 216 permutacions, però no sempre és possible).")
  (prompt   "\n\nF3: Permutació de tríades de Files (6 permutacions).")
  (prompt "\n\n\nFS: Simetria respecte a la Fila central (es pot fer amb tres ")
  (prompt "F1 i una F3).")
  (prompt "\n\n\nC1: Permutació de Columnes d'una mateixa tríada\n    (6 ")
  (prompt "permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).")
  (prompt "  \n\nC2: Permutació de Columnes de tríades diferents")
  (prompt "\n    (fins a 216 permutacions, però no sempre és possible).")
  (prompt   "\n\nC3: Permutació de tríades de Columnes (6 permutacions).")
  (prompt "\n\n\nCS: Simetria respecte a la Columna central (es pot fer amb tres")
  (prompt " C1 i una C3).")
  (prompt "\n\n\nTR: Transposició de files i columnes.")
  (prompt "\n\n\nPV: Permutació entre Valors 1 a 9 estesa a tot el SUDOKU (tot i")
  (prompt " que muntat com\n    una seqüència de permutacions binàries, hi ha 9!")
  (prompt " = 362.880 possibilitats).\n\n")
  (SEGUIR ()))

(defun FES-+9*9+- (/ -+9*9+- E F) ; Si (= ABC "A"), posa signe "-" als valors del
  (setq -+9*9+- **9*9**)          ; SUDOKU-PROBLEMA en CANON->VARIANT, per tal que
  (foreach P PPM                  ; (GRAF-9*9 T) els destaqui en blanc sobre gris
    (setq X (fix (/ (+ (caar P)  (caadr P))  2)) ; (passen a la capa NORM-1 mentre
          Y (fix (/ (+ (cadar P) (cadadr P)) 2)) ; els altres valors de **9*9** es
          F (nth Y -+9*9+-) E (nth X F)          ; queden a la capa NORM-0).
          -+9*9+- (subst (subst (- E) E F) F -+9*9+-))))

(defun AREES (/ G)
  (setq K -1)
  (if (wcmatch E "*column*")
      (if (< F 3)
          (repeat 9 (setq K (1+ K)
                          G (list (mapcar '- (list K 0) TG)
                                  (mapcar '+ (list K 8) TG))
                          GG (cons G GG)))
          (repeat 3 (setq K (1+ K)
                          G (list (mapcar '- (list (* 3 K) 0) TG)
                                  (mapcar '+ (list (+ 2 (* 3 K)) 8) TG))
                          GG (cons G GG))))
      (if (< F 3)
          (repeat 9 (setq K (1+ K)
                          G (list (mapcar '- (list 0 K) TG)
                                  (mapcar '+ (list 8 K) TG))
                          GG (cons G GG)))
          (repeat 3 (setq K (1+ K)
                          G (list (mapcar '- (list 0 (* 3 K)) TG)
                                  (mapcar '+ (list 8 (+ 2 (* 3 K))) TG))
                          GG (cons G GG)))))
  (setq GG (reverse GG)))

(defun PREVIES ()
  (setq E (if (= (substr RESP 1 1) "F") "fil" "column")
        F (atoi (substr RESP 2 1))
        E (if (= F 1) (strcat E "a") (if (= F 2) E (strcat "tríada de " E "es"))))
  (AREES))

(defun ZONA (P)
  (setq Z () K -1)
  (foreach G GG
    (if (and (not Z) (setq K (1+ K))
             (< (caar G) (car P))  (< (cadar G) (cadr P))
             (< (car P) (caadr G)) (< (cadr P) (cadadr G)))
        (setq Z K)))
  Z)
```

```
(defun CONTROL (/ OK)
  (if ZZ
      (if (= Z (car ZZ))
          (setq OK ())
          (if (= F 3)
              (setq OK T)
              (progn
                (setq OK (= F 2))
                (if (or (= Z I)
                        (= Z (1+ I))
                        (= Z (+ 2 I)))
                    (setq OK (= F 1))))))
      (setq OK T I (if (< F 3) (* 3 (/ Z 3)) 0)))
  OK)

(defun INPUNT (MSG / GR Z)
  (prompt MSG)
  (while (not (and (= (car (setq GR (grread))) 3)
                   (ZONA (cadr GR))
                   (CONTROL)))
         (if Z (prompt (strcat "\nNo s'hi val>" (substr MSG 2)))))
  (setq ZZ (cons Z ZZ)))

(defun PERMUT-G (/ L)
  (setq L (mapcar '(lambda (Z) (- Z I)) ZZ))
  (cond ((or (equal L '(1 2 1)) (equal L '(2 1 2))) '(0 2 1))
        ((or (equal L '(0 1 0)) (equal L '(1 0 1))) '(1 0 2))
        ((or (equal L '(0 1 2)) (equal L '(1 2 0)) (equal L '(2 0 1))) '(1 2 0))
        ((or (equal L '(0 2 1)) (equal L '(1 0 2)) (equal L '(2 1 0))) '(2 0 1))
        ((or (equal L '(0 2 0)) (equal L '(2 0 2))) '(2 1 0))))

(defun TR-9**9 (F/C)
  (setq **9**9** (if (= (substr RESP 1 1) (if F/C "F" "C"))
                     **9**9**
                     (TRANSPOSAR **9**9**))))

(defun CAN->VAR (/ 9**9)
  (TR-9**9 T)
  (if (= F 1)
      (progn
        (setq 9**9 **9**9**)
        (foreach J '(0 1 2)
          (setq K (+ I J) **9**9** (subst (nth J JJ) (nth K 9**9) **9**9**)))
        (foreach J '(0 1 2)
          (setq K (+ I J) **9**9** (subst (nth K 9**9) J **9**9**))))
      (setq **9**9** (F=3 **9**9**)))
  (TR-9**9 T))

(defun EXECUTA (FUN / 9*9)
  (if (= ABC "A")
      (progn
        (setq 9*9 **9**9** **9**9** **9*9**)
        (FUN)
        (setq **9*9** **9**9** **9**9** 9*9)))
  (FUN))

(defun F/C (/ E F GG JJ ZZ)
  (PREVIES)
  (while (not (setq ZZ ()
                    JJ (INPUNT (strcat T1 E T2))
                    JJ (INPUNT T3)
                    JJ (INPUNT (strcat T4 E T5))
                    JJ (PERMUT-G)))
         (prompt "\nNo has definit cap permutació. TORNA-HI!"))
  (EXECUTA CAN->VAR))
```

```
(defun EXPLORA (/ EFD FDE HIG IGH LLL)
  (TR-9**9 T)
  (setq LL ())
  (foreach I '(0 1 2 3 4 5)
    (setq L (nth I **9**9**)
          EFD (list (nth 4 L) (nth 5 L) (nth 3 L))
          FDE (list (nth 5 L) (nth 3 L) (nth 4 L))
          HIG (list (nth 7 L) (nth 8 L) (nth 6 L))
          IGH (list (nth 8 L) (nth 6 L) (nth 7 L))
          LLL (list (append EFD HIG) (append EFD IGH)
                    (append FDE HIG) (append FDE IGH)))
    (foreach J '(3 4 5 6 7 8)
      (if (>= J (* (1+ (/ I 3)) 3))
          (progn
            (setq K (nth J **9**9**))
            (if (member (member (nth 3 K) K) LLL)
                (setq LL (cons (list I J) LL)))))))
  (TR-9**9 T)
  (setq LL (reverse LL)))

(defun PERM-BIN ()
  (TR-9**9 T)
  (setq J (nth (car ZZ) **9**9**)
        K (nth (cadr ZZ) **9**9**)
        **9**9** (subst () J **9**9**)
        **9**9** (subst J  K **9**9**)
        **9**9** (subst K () **9**9**))
  (TR-9**9 T))

(defun F//C (/ E F GG JJ ZZ)
  (PREVIES)
  (prompt "\n.\n.\nAquesta VARIANT ESCOLLIDA ")
  (if (EXPLORA)
      (progn
        (prompt (strcat "permet permutar les " E "es"))
        (foreach L LL
          (prompt (strcat " " (itoa (1+ (car L))) "ª i " (itoa (1+ (cadr L))) "ª"
                          (if (equal L (last LL))
                                       "."
                                       (if (equal L (cadr (reverse LL)))
                                           ", i" ",")))))
        (while (not (setq ZZ () ZZ (INPUNT (strcat T1 "primera " E "a: "))
                          ZZ (INPUNT (strcat T1 "segona "  E "a: "))
                          ZZ (if (< (car ZZ) (cadr ZZ))
                          ZZ (reverse ZZ))
                          JJ (member ZZ LL)))
               (prompt (strcat "\nAquesta permutació (" (itoa (1+ (car ZZ))) "-"
                               (itoa (1+ (cadr ZZ))) ") no és permesa. TORNA-HI!")))
        (EXECUTA PERM-BIN))
      (prompt (strcat "no permet permutar " E "es de tríades diferents.\n"))))

(defun SIM ()
  (TR-9**9 ())
  (foreach F **9**9** (setq **9**9** (subst (reverse F) F **9**9**)))
  (TR-9**9 ()))

(defun INVALOR (MSG) (prompt MSG) (setq N ()) (TECLA-P T) N)

(defun 2-9V (/ CURSOR MASC N1 N2 N-F N-9*9)
  (TECLAT)
  (command "BORRA" CURSOR "")
  (setq MASC L1A9)
  (while (INVALOR (strcat "\n.\n." T6 "a substituir" (if N1 " <sortir>" "") ": "))
         (setq N1 (nth (1- N) MASC)
               N2 (nth (1- (INVALOR (strcat T6 "que el substituirà: "))) MASC))
         (if (/= N1 N2)
             (progn
               (command "BORRA" SS "")
```

```
            (setq MASC (subst N2 0 (subst N1 N2 (subst 0 N1 MASC))))
            (MASCARA MASC)
            (foreach 9*9 (if (= ABC "A")
                             (list **9*9** **9**9**)
                             (list **9**9**))
              (setq N-9*9 ())
              (foreach F 9*9
                (setq N-F ())
                (foreach E F
                  (setq I (/ E (abs E))
                        N-F (cons (if (= (abs E) N1)
                                      (* N2 I)
                                      (if (= (abs E) N2) (* N1 I) E))
                                  N-F)))
                (setq N-9*9 (cons (reverse N-F) N-9*9)))
              (setq N-9*9 (reverse N-9*9))
              (if (= ABC "A")
                  (if (equal 9*9 **9*9**)       ; Per a quan el primer N-9*9
                      (if (equal 9*9 **9**9**)  ; (nou **9*9**) sigui igual
                          (setq **9**9** N-9*9) ; al **9**9** original. També
                          (setq **9*9** N-9*9)) ; per a quan els dos originals
                      (setq **9**9** N-9*9))    ; siguin iguals.
                  (setq **9**9** N-9*9)))
            (command "ESPACIOM")
            (GRAF-9*9 (= ABC "A"))
            (command "ESPACIOP"))))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" "" "ESPACIOM"))

(defun CANON->VARIANT (/ T1 T2 T3 T4 T5 T6 RESP OK)
  (graphscr)
  (setq T1 "\n   Assenyaleu (dreta) la " T2 " que vulgueu moure: "
        T3 "\n   Assenyaleu (dreta) on voleu situar-la: "
        T4 "\n   Assenyaleu (dreta) on voleu situar la "
        T5 " desplaçada: "
        T6 "\n   CLIC (centre) sobre el nombre " RESP "")
  (while RESP
         (if (/= (substr (getvar "LASTPROMPT") 1 35)
                 "Aquesta VARIANT ESCOLLIDA no permet")
             (prompt "\n."))
         (initget "F1 F2 F3 FS C1 C2 C3 CS TR PV")
         (setq RESP (getkword (strcat "\n.\nTipus de permutació: F1/F2/F3/FS/"
                                      "C1/C2/C3/CS/TR/PV <sortir>: ")))
         (if (not OK)
             (progn
               (if (= ABC "A")
                   (progn
                     (RETOL-1 '(-0.75 -0.55) "V. E. & SUDOKU INICIAL")
                     (setq **9*9** (FES-+9*9+-))))
               (setq OK T)
               (command "ESPACIOM")
               (if (= ABC "A") (GRAF-9*9 T))))
         (if (or (= RESP "F1") (= RESP "F3") (= RESP "C1") (= RESP "C3"))
             (F/C)
             (if (or (= RESP "F2") (= RESP "C2"))
                 (F//C)
                 (if (or (= RESP "FS") (= RESP "CS"))
                     (EXECUTA SIM)
                     (if (= RESP "TR")
                         (setq **9*9** (if (= ABC "A")
                                           (TRANSPOSAR **9*9**) **9*9**)
                               **9**9** (TRANSPOSAR **9**9**))
                         (if (= RESP "PV") (2-9V))))))
         (GRAF-9*9 (= ABC "A")))
  (initget "Si No")
  (setq CONF
        (getkword "\n.\n.\nUs sembla bé com a solució de treball (S/N)? <S>: ")
        CONF (if CONF CONF "Si")))
```

```
(defun TRIA-METODE ()
  (textscr)
  (prompt "\n\n\n\n\nDisposeu de dos mètodes per obtenir un SUDOKU-PROBLEMA ")
  (prompt "d'una SUDOKU-SOLUCIÓ\n(aquesta VARIANT ESCOLLIDA) reactivant caselles")
  (prompt " (deixant-les visibles):")
  (prompt "\n\n1) Anar-ho fent fins haver activat tota la solució, moment en què")
  (prompt " el programa\n   remuntarà el procés per només deixar visibles les ")
  (prompt "caselles imprescindibles\n   (a partir d'un cert punt, l'usuari haurà")
  (prompt " d'esperar autorització per activar\n   cada nova casella, però els ")
  (prompt "temps d'espera cada vegada seran més breus).")
  (prompt "\n\n2) Anar-ho fent fins que el programa ens informi que només hi ha ")
  (prompt "una solució\n   compatible amb les caselles activades ")
  (prompt "(SUDOKU-PROBLEMA): la SUDOKU-SOLUCIÓ\n   (entremig del procés es ")
  (prompt "produiran dues pauses, que poden ser ben llargues,\n   però tret ")
  (prompt "d'això l'activació de cada casella -revisable- serà immediata).")
  (initget "1 2")
  (setq 1-2 (getkword "\n\nTrieu l'opció 1 o 2 <2>: ") 1-2 (if 1-2 1-2 "2")))

(defun ORDENA-3 ()
  (setq A (caar L) B (caadr L) C (caaddr L) D (assoc (max A B C) L)
        A (assoc (min A B C) L) L (subst () D (subst () A L))))

 (defun VAR->NORM (/ K L 9**9 A B C D)
  (setq K -1)
  (repeat 3
    (setq L ())
    (repeat 3 (setq K (1+ K) L (cons (nth K **9**9**) L)))
    (ORDENA-3)
    (foreach E L (if E (setq L (list A E D))))
    (setq 9**9 (append 9**9 (list L))))
  (setq K -1 L ())
  (foreach E 9**9 (setq L (cons (cons (caar E) E) L)))
  (ORDENA-3)
  (foreach E L (if E (setq **9*9** (append (cdr A) (cdr E) (cdr D))))))

(defun PERMUT-N (A)
  (if (atom A)
      (cond ((= A 0)  '(0 1 2))
            ((= A 1)  '(0 2 1))
            ((= A 2)  '(1 0 2))
            ((= A 3)  '(1 2 0))
            ((= A 4)  '(2 0 1))
            (   T     '(2 1 0)))
      (cond ((equal A '(0 1 2)) 0)
            ((equal A '(0 2 1)) 1)
            ((equal A '(1 0 2)) 2)
            ((equal A '(2 0 1)) 3)
            ((equal A '(1 2 0)) 4)
            (   T   5))))

(defun PRE-NORM->VAR (/ L* L** N F F1 F2 F3 F4)
  (foreach L  (reverse **9*9**) (setq L*  (cons (car L) L*)))
  (foreach L (reverse **9**9**) (setq L** (cons (car L) L**)))
  (setq N1 (- 9 (length (member (nth 0 L*) L**))) F1 (/ N1 3)
        N2 (- 9 (length (member (nth 3 L*) L**))) F2 (/ N2 3)
        N3 (- 9 (length (member (nth 6 L*) L**))) F3 (/ N3 3)
        N4 (list F1 F2 F3) N4 (PERMUT-N N4) F1 (rem N1 3)
        F2 (- 9 (length (member (nth 1 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 2 L*) L**))) F3 (rem F3 3)
        N1 (list F1 F2 F3) N1 (PERMUT-N N1) F1 (rem N2 3)
        F2 (- 9 (length (member (nth 4 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 5 L*) L**))) F3 (rem F3 3)
        N2 (list F1 F2 F3) N2 (PERMUT-N N2) F1 (rem N3 3)
        F2 (- 9 (length (member (nth 7 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 8 L*) L**))) F3 (rem F3 3)
        N3 (list F1 F2 F3) N3 (PERMUT-N N3)))
```

```
(defun NORMTEXT-9*9 ()
  (prompt "\n\n\nSi anomenem SOLUCIÓ NORMALITZADA aquella en què les files de ")
  (prompt "cada tríada\nestan ordenades numèricament d'avall cap amunt, i les ")
  (prompt "tríades de files també\nestan ordenades atenent la fila inferior, ")
  (prompt "a la finestra de l'esquerra veureu\nla V. E. NORMALITZADA, referida ")
  (prompt "a la VARIANT ESCOLLIDA que resta a la dreta\n(per obtenir-la el ")
  (prompt "programa haurà fet fins a tres permutacions F1 i una F3).")
  (prompt "\n\nJugant amb les solucions normalitzades serà força més ràpid el ")
  (prompt "processat per\ndetectar amb quantes caselles plenes la SUDOKU-SOLUCIÓ")
  (prompt " ja queda determinada.")
  (SEGUIR ())
  (VAR->NORM)
  (PRE-NORM->VAR)
  (graphscr)
  (GRAF-9*9 ())
  (command "ESPACIOP")
  (RETOL-1 '(-0.75 -0.55) "V. E. NORMALITZADA")
  (prompt "\n\nCal tenir l'arxiu 123456789.txt a la ruta de cerca d'AutoCAD. ")
  (prompt "Si no l'hi teniu,\npodeu copiar-lo en el directori d'aquest dibuix ")
  (prompt "i després seguir com si no res.")
  (SEGUIR ()))

(defun EXPLICACIO ()
  (textscr)
  (prompt "\n\n\n\nPer construir el SUDOKU-PROBLEMA a partir d'aquest ")
  (prompt "SUDOKU-SOLUCIÓ, haureu de\nseguir els passos següents:")
  (prompt "\n\n1) Com que els valors quedaran velats (en gris), haureu d'anar ")
  (prompt "fent clic a les\n   caselles que vulgueu convertir en públiques ")
  (prompt "(visibles per al jugador). Els\n   clics poden fer-se sobre la ")
  (prompt "V. E. NORMALITZADA o sobre la VARIANT ESCOLLIDA\n   (esquerra o ")
  (prompt "dreta) però per passar de l'una a l'altra caldrà un altre clic.")
  (prompt "\n\n2) Les caselles públiques quedaran destacades (blanc/negre) ")
  (prompt "en ambdues bandes,\n   però, si hi torneu a fer un clic, ")
  (prompt "passaran de nou a ser secretes (en gris).")
  (prompt "\n\n3) Quan el conjunt de caselles públiques assoleixi les ")
  (prompt "condicions necessàries\n   per a una solució única (haver-n'hi ")
  (prompt "almenys 17; que cap tríada de files no\n   en tingui 2 de buides; ")
  (prompt "que cap tríada de columnes no en tingui 2 de buides)\n   hi haurà ")
  (prompt "la pausa més llarga. El text inferior mostrarà que hi ha activitat")
  (prompt "\n   i, quan aquesta cessi, podreu llegir-ne el resum i saber ")
  (prompt "quantes SOLUCIONS\n   NORMALITZADES, compatibles amb la ")
  (prompt "V. E. NORMALITZADA, s'han detectat.")
  (prompt "\n\n4) Si n'hi ha més d'una, haureu de seguir fent clic ")
  (prompt "fins a tenir-ne només una.")
  (prompt "\n\n5) Una nota us advertirà que el fet que sols quedi una solució ")
  (prompt "normalitzada no\n   exclou que hi hagi més SOLUCIONS, ")
  (prompt "NO NORMALITZADES però també compatibles,\n   i anunciarà una segona ")
  (prompt "pausa per calcular i fer-vos saber quantes n'hi ha.")
  (prompt "\n\n6) Si hi ha més d'una solució, caldrà seguir fent clic fins que ")
  (prompt "només en quedi\n   una, que serà la normalitzada.\n")
  (SEGUIR ()))

(defun YIN-YANG ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "ARCO" "0,-0.15" "0.15,0" "0,0.15" "ARCO" "" "0,-0.15"
           "ARCO" "0,0" "0.075,0.075" "0,0.15"
           "ARCO" "0,0" "-0.075,-0.075" "0,-0.15"
           "CIRCULO" "0,0.075" 0.015 "CIRCULO" "0,-0.075" 0.02
           "COLOR" G  "SOMBREA" "S" "-0.15,0" "0.15,0" ""
           "BLOQUE" "C" "0,0" "LT" ""
           "SOMBREA" "S" "0,0.5" "-0.15,0" "0.15,0" "" "BORRA" "LT" ""
           "COLOR" 7 "SOMBREA" "S" "0.15,0" "-0.075,-0.075" "0.075,0.075"
                                   "0,-0.055" "0,0.06" ""
           "UY" "DESIGNA" "P" "LT" ""
           "BORRA" "C" "-0.2,0.6" "0.2,-0.6" "E" "P" ""
           "DESIGNA" "LT" "" "ESPACIOM"))
```

```
(defun COMMUTA (LL)
  (subst (subst (if V (list X Y) N) (if V N (list X Y)) (nth Y LL))
         (nth Y LL) LL))

(defun V+- (C)
  (if V
      (command "BORRA" (ssname SS 0) "")
      (command "COPIA" SS "" "0,0" ""
               "CAMBPROP" "LT" "" "C" C "")))

(defun Y1 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 0) N1) ((= Y 3) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (+ Y K)
              (if (or (= N K3) (= N K4))
                  (+ Y (* 2 K))
                  Y))))

(defun Y2 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 1) N1) ((= Y 4) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (- Y K)
              (if (or (= N K3) (= N K4))
                  (+ Y K)
                  Y))))

(defun Y3 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 2) N1) ((= Y 5) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (- Y K)
              (if (or (= N K3) (= N K4))
                  (- Y (* 2 K))
                  Y))))

(defun PUNT () (setq P (list X Y) PX (mapcar '- P TG) PY (mapcar '+ P TG)))

(defun V->N (/ N)
  (cond ((< Y 3) (Y1 3 2 3 4 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 4 1 3))
        (T (Y3 3 1 4 3 5)))
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 3 4 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 4 1 3))
        (T (Y3 1 1 4 3 5)))
  (PUNT))

(defun N->V (/ N)
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 4 3 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 3 1 4))
        (T (Y3 1 1 3 4 5)))
  (cond ((< Y 3) (Y1 3 2 4 3 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 3 1 4))
        (T (Y3 3 1 3 4 5)))
  (PUNT))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if (and (= 1-2 "2") Q) (setq P Q X (car P) Y (cadr P) PX QX PY QY))
        (setq N (ELEMENT **9**9**)
              V (atom (ELEMENT **V**V**))
              **V**V** (if (and (= 1-2 "1") (not V)) **V**V** (COMMUTA **V**V**))
              SS (ssget "_C" PX PY))
        (if (and (= 1-2 "2") (not Q)) (setq Q P QX PX QY PY))
        (V+- "VAR-1")
        (if (= 1-2 "2")
            (progn
              (command "CVPORT" 2)
              (V->N)
```

```
              (setq **V*V** (COMMUTA **V*V**)
                    SS (ssget "_C" PX PY))
              (V+- "NORM-1")
              (command "CVPORT" 3))))
      (progn
        (setq N (ELEMENT **9*9**)
              V (atom (ELEMENT **V*V**))
              **V*V** (COMMUTA **V*V**)
              SS (ssget "_C" PX PY) Q P QX PX QY PY)
        (V+- "NORM-1")
        (command "CVPORT" 3)
        (N->V)
        (setq **V**V** (COMMUTA **V**V**)
              SS (ssget "_C" PX PY))
        (V+- "VAR-1")
        (command "CVPORT" 2)
        (setq P Q X (car P) Y (cadr P) PX QX PY QY Q ())))))

(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM) PY (cadar PPM)
            X (fix (/ (+ (car PX) (car PY)) 2))
            Y (fix (/ (+ (cadr PX) (cadr PY)) 2)) P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (and (not POST) (> (last (car PPM)) 0))
            (setq NOPLUS (COMMUTA NOPLUS)))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (setq M 1 GR N)
        (if (not LLL) (INI-LLL))
        (if (not POST) (SUD-MIN NOPLUS T))
        (MASCARA (ELEMENT LLL))
        (setq N GR PPM (cons (list PX PY M) PPM)
              LLL (ACTUALITZA **V**V** N P LLL T)
              **V**V** (car LLL) LLL (cadr LLL)
              LLL/LLL (cons LLL LLL/LLL)))))

(defun NUMCOMPAT (P N NOU / PS)
  (setq K 0 KK 0)
  (foreach SUD SUDOKUS-V*V
    (setq K (1+ K)
          OK (if NOU (= (nth (car P) (nth (cadr P) SUD)) N) T)
          PS (if NOU (cons OK PS)))
    (if OK (progn
             (foreach S S-V*V (if (and OK (not (nth K S))) (setq OK ())))
             (if OK (setq KK (1+ KK))))))
  (if NOU (setq S-V*V (cons (cons P (reverse PS)) S-V*V))))

 (defun C:CARREGA-123456789 (/ A)
  (setq RUTA (findfile "123456789.txt") A (open RUTA "r") L1 ())
  (repeat 362880 (setq L1 (cons (read-line A) L1)))
  (setq L1 (reverse L1))
  (close A))

(defun FOTO-V*V (/ FOTO)
  (setq J -1)
  (foreach L **V*V**
    (setq I -1 J (1+ J))
    (foreach E L
      (setq I (1+ I))
      (if (atom E) (setq FOTO (cons (list I J) FOTO)))))
  FOTO)

(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V)
  (if (not L1) (C:CARREGA-123456789))
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
```

```
 (repeat 3
   (setq J (- J 9) K (+ K 3))
   (repeat 3
     (setq J (+ J 3) K (- K 3) W ())
     (repeat 3
       (setq J (- J 3) K (1+ K))
       (repeat 3
         (setq J (1+ J) V (nth J (nth K **V*V**)))
         (if (atom V) (setq W (cons V W)))))
     (cond ((= J 2) (setq A W))
           ((= J 5) (setq B W))
           (T       (setq C W))))
   (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
 (setq L1C (reverse L1C) J -1)
 (repeat 9
   (setq J (1+ J) K -1 L ())
   (repeat 9 (setq K (1+ K) C (nth J (nth K **V*V**))
                   L (if (atom C) (cons C L) L)))
   (setq L2C (cons L L2C)))
 (setq L2C (reverse L2C) LINIES-V*V () K -1)
 (terpri)
 (repeat 9
   (setq LINIA-V*V () L "" K (1+ K))
   (mapcar '(lambda (E F G)
              (setq L (strcat L (if (or F G)
                                    (if (atom E)
                                        (itoa E)
                                        (progn
                                          (setq C "]")
                                          (foreach H (append F G)
                                            (setq C (strcat (itoa H) C)))
                                          (strcat "[~" C)))
                                    "#"))))
           (nth K **V*V**) (nth K L1C) L2C)
   (cond ((= K 0) (setq A "123456789" B "198765432"))
         ((= K 1) (setq A "213456789" B "897654321"))
         ((= K 2) (setq A "312456789" B "987654321"))
         ((= K 3) (setq A "213456789" B "498765321"))
         ((= K 4) (setq A "312456789" B "897654321"))
         ((= K 5) (setq A "412356789" B "987654321"))
         ((= K 6) (setq A "312456789" B "798654321"))
         ((= K 7) (setq A "412356789" B "897654321"))
         (T       (setq A "512346789" B "987654321")))
   (foreach E (reverse (member B (reverse (member A L1))))
     (if (wcmatch E L) (setq LINIA-V*V (cons E LINIA-V*V))))
   (setq LINIES-V*V (cons (reverse LINIA-V*V) LINIES-V*V)
         *K* (itoa (length LINIA-V*V)) *KK* (cons *K* *KK*))
   (terpri)
   (prompt (strcat "\nEntre 362880 línies, de compatibles amb la " (itoa (1+ K))
                   "ª de la V. E. NORM. n'hi ha " *K* "."))
 (setq LINIES-V*V (reverse LINIES-V*V) L1 () *KK* (reverse *KK*))
 (terpri))

(defun COMPAT-2L ()
 (setq K 1 OK T)
 (while (and OK (< K 9))
        (setq K (1+ K) OK (/= (substr T0 K 1) (substr T1 K 1)))))

(defun COMPAT-3L (/ M)
 (setq K 1 OK T)
 (while (and OK (< K 9))
        (setq K (1+ K) OK (/= (substr T1 K 1) (substr T2 K 1) (substr T0 K 1))))
 (if OK
     (progn
       (setq K -2)
       (repeat 2 (if OK
                     (progn
                       (setq M "" K (+ 3 K))
```

```
                        (foreach L (list T0 T1 T2)
                          (setq M (strcat M (substr L K 3))))
                        (foreach N TOT
                          (if (and OK (not (wcmatch M N))) (setq OK ()))))))))
  OK)

(defun ACTLOC () (setq LOC (cons (cons A K) LOC)))

(defun LOCT (TT / A Z LOC)
  (setq A (substr (caar TT) 1 1) Z (substr (car (last TT)) 1 1) K 0)
  (ACTLOC)
  (while (< A Z)
         (setq A (itoa (1+ (atoi A))))
         (while (< (car (nth (setq K (1+ K)) TT)) A))
         (setq A (substr (car (nth K TT)) 1 1))
         (ACTLOC))
  (reverse LOC))

(defun COLS-3L (I / C CC)
  (setq K 10)
  (repeat 9
    (setq K (1- K) C "")
    (foreach J I (setq C (strcat C (substr J K 1))))
    (setq CC (cons C CC))))

(defun PERF-COLS (CC) (mapcar '(lambda (C) (strcat "*[" C "]*")) CC))

(defun SUPR (E L)
  (append (reverse (cdr (member E (reverse L)))) (cdr (member E L))))

(defun SUD-NUM (/ LN LLN)
  (foreach 3L (list T2 T1 T0)
    (foreach L (reverse 3L)
      (setq LN () K 10)
      (repeat 9 (setq K (1- K) LN (cons (atoi (substr L K 1)) LN)))
      (setq LLN (cons LN LLN)))))

(defun MEMBRE (K L) (repeat K (setq L (cdr L))) L)

(defun COMPAT-PERFILS (/ LL0 LL1 L2 LL2 3L M N TERCETS-V*V CC0 CC1 CC2 CCC1 CCC2
                         TT0 TT1 TT2 LOCT1 LOCT2 TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  (repeat 9 (setq K (1+ K) TOT (cons (strcat "*" (itoa K) "*") TOT)))
  (repeat 3
    (setq LL0 (car LINIES-V*V) LL1 (cadr LINIES-V*V) LL2 (caddr LINIES-V*V)
          LINIES-V*V (cdddr LINIES-V*V) TERCETS-V*V ())
    (foreach T0 LL0
      (foreach T1 LL1
        (if (and (< T0 T1) (COMPAT-2L))
            (foreach T2 LL2
              (if (and (< T1 T2) (COMPAT-3L))
                  (setq TERCETS-V*V (cons (list T0 T1 T2) TERCETS-V*V)))))))
    (if TERCETS-V*V
        (setq *T* (itoa (length TERCETS-V*V)) *TT* (append *TT* (list *T*))
              SUDOKUS-V*V (append SUDOKUS-V*V (list (reverse TERCETS-V*V)))))
    (prompt (strcat "\nEntre " (nth *K* *KK*) " línies compatibles amb la "
                    (itoa (setq *K* (1+ *K*))) "ª, " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª i " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª,\nhi ha " *T* " tríades "
                    "normalitzades compatibles amb la V. E. NORMALITZADA.\n")))
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
```

```
        (foreach T0 TT0
          (if (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0)
                    L1 (SUPR (substr (cadr T0) 1 1) TOT)
                    L1 (SUPR (substr (caddr T0) 1 1) L1)
                    PRIM1 (cdr (assoc (car L1) LOCT1)))
              (mapcar '(lambda (T1 CC1)
                         (setq L2 (cdr L1) K -1 OK T)
                         (while (and OK (< (setq K (1+ K)) 9))
                                (if (wcmatch (nth K CC1) (nth K CC0))
                                    (setq OK ())))
                         (if (and OK (setq CC1 (PERF-COLS CC1)
                                           L2 (SUPR (substr (cadr T1) 1 1) L2)
                                           L2 (SUPR (substr (caddr T1) 1 1) L2)
                                           PRIM2 (cdr (assoc (car L2) LOCT2))))
                             (mapcar '(lambda (T2 CC2)
                                        (setq OK T K -1)
                                        (while (and OK (< (setq K (1+ K)) 9))
                                               (if (or (wcmatch
                                                         (setq J (nth K CC2))
                                                         (nth K CC0))
                                                       (wcmatch J (nth K CC1)))
                                                   (setq OK ())))
                                        (if OK (setq SUDOKUS-V*V
                                                     (cons (SUD-NUM) SUDOKUS-V*V)
                                                     S-V*V (cons T S-V*V))))
                                     (member (nth PRIM2 TT2) TT2)
                                     (MEMBRE PRIM2 CCC2))))
                    (member (nth PRIM1 TT1) TT1) (MEMBRE PRIM1 CCC1))))
        (setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
              KK (length S-V*V) S-V*V (list (cons P S-V*V)))
        (prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                                       (nth 1 *TT*) " tríades 4-5-6 i les "
                                       (nth 2 *TT*) " tríades 7-8-9,"
                        "\nde solucions normalitzades compatibles amb la V. E. "
                        "NORMALITZADA n'hi ha " (itoa KK) "."))
        (textscr)
        (SEGUIR ())
        (graphscr))))

(defun RE+MEMBER (R-9*9 R-LLL / 2R R-N R-P F C0 C3 C6)
  (if (COMPLET-9*9 R-9*9)
      (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9) SUDOKUS-V*V)
            KKK (cons T KKK))
      (progn
        (if (and PRE (> (cadr R-P) 0))
            (setq PRE () F (nth 0 R-9*9)
                  C0 (nth 0 F) C3 (nth 3 F) C6 (nth 6 F)
                  NORM (and (< C0 C3 C6)
                            (< C0 (nth 1 F) (nth 2 F))
                            (< C3 (nth 4 F) (nth 5 F))
                            (< C6 (nth 7 F) (nth 8 F)))))
        (if (or PRE (not NORM))
            (progn
              (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
              (foreach E L1A9
                (if (member E R-N)
                    (progn
                      (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                      (RE+MEMBER (car 2R) (cadr 2R))))))
            (if NORM (setq PRE T))))))

(defun ANORMALS (NORM=1 / L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P PRE NORM)
  (setq L0A8 '(0 1 2 3 4 5 6 7 8) FOTO1 () FOTO2 (FOTO-V*V))
  (if (not NORM=1) (setq KKK () SUDOKUS-V*V ()))
  (foreach X L0A8
    (setq XX ())
```

```
    (foreach Y L0A8
      (if (atom (setq V (ELEMENT **V*V**))) (setq XX (cons V XX))))
    (setq XXX (cons XX XXX)))
  (setq XXX (reverse XXX))
  (foreach Y L0A8
    (setq YY ())
    (foreach X L0A8
      (if (atom (setq V (ELEMENT **V*V**))) (setq YY (cons V YY))))
    (setq YYY (cons YY YYY)))
  (setq YYY (reverse YYY))
  (foreach YY '(0 3 6)
    (foreach XX '(0 3 6)
      (setq L () Y -1)
      (foreach VV **V*V**
        (setq Y (1+ Y) X -1)
        (if (not (or (< Y YY) (> Y (+ YY 2))))
            (progn
              (foreach V VV
                (setq X (1+ X))
                (if (not (or (< X XX) (> X (+ XX 2))))
                    (if (atom V) (setq L (cons V L))))))))
      (setq XY (cons L XY))))
  (setq XY (reverse XY))
  (foreach Y L0A8
    (setq LL ())
    (foreach X L0A8
      (setq L ())
      (if (listp (ELEMENT **V*V**))
          (foreach E L1A9
            (setq L (cons (if (or (member E (nth X XXX)) (member E (nth Y YYY))
                                  (member E (nth (+ (/ X 3) (* (/ Y 3) 3)) XY)))
                              () E)
                          L))))
      (setq LL (cons (reverse L) LL)))
    (setq LL (reverse LL) LLL (cons LL LLL)))
  (setq LLL (reverse LLL))
  (if NORM=1
      (progn
        (setq PRE T LL ())
        (foreach EE (TRANSPOSAR **V*V**)
          (setq L ())
          (foreach E EE (setq L (cons (if (atom E) E (reverse E)) L)))
          (setq LL (cons (reverse L) LL)))
        (setq LL (reverse LL))
        (RE+MEMBER LL (TRANSPOSAR LLL)))
      (RE+MEMBER **V*V** LLL))
  (setq KK (length KKK) KKK (cons P KKK) S-V*V (list KKK))
  (if (and NORM=1 (= KK 1)) (setq KKK ())) ; La solució única és la normalitzada.

(defun FES-SUDOKU-2 (/ LINIES-V*V *K* *KK*)
  (setq Q ())
  (CLIC)
  (if POST
      (if V
          (if (or (member P FOTO1) (member P FOTO2))
              (progn
                (command "ESPACIOP" "BORRA" "LT" ""
                         "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                (RETOL-2 "Anul-lant" "aquesta" "casella," "caldrà" "recalcular"
                         "solucions" "compatibles" "i tornar a" "esperar...")
                (if FOTO2
                    (progn
                      (SEGUIR T)
                      (command "DESHACER" "R" "DESHACER" "M"
                               "INSERT" "C" "0,0" "" "" "")
                      (RETOL-2 "...i, si surt" "\"EL SUDOKU" "JA TÉ UNA" "SOLUCIÓ"
                               "ÚNICA\", mai" "no sabreu" "si sobren" "caselles"
                               "plenes."")))
```

```
              (initget "Si No")
              (setq CONF (getkword "\n.\n.\nConfirmeu l'anul·lació (S/N)? <N>:")
                    CONF (if CONF CONF "No"))
              (if (= CONF "Si")
                  (setq POST ())
                  (progn (command "DESHACER" "R" "UY" "ESPACIOM") (CLIC))))
            (progn
              (setq I (assoc P S-V*V)
                    J (reverse (cdr (member I (reverse S-V*V))))
                    S-V*V (append J (cdr (member I S-V*V)))
                    I (caar S-V*V) J (nth (car I) (nth (cadr I) **9*9**)))
              (NUMCOMPAT I J ())))
          (NUMCOMPAT P N T)))
(if (not POST)
    (progn
      (if (and V (or (member P FOTO1) (member P FOTO2)))
          (setq POST T)
          (if (not V) (SUD-MIN **V*V** ())))
      (if POST
          (progn
            (if V
                (command "DESHACER" "R")
                (command "ESPACIOP" "BORRA" "LT" ""))
            (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
            (if (or (not (or FOTO1 FOTO2))
                    (member P FOTO1))
                (progn ; No hi ha FOTO1 (ni FOTO2) però s'ha de crear
                       ;                perquè SUD-MIN ha activat POST, o bé
                       ; ja hi ha FOTO1 però cal actualitzar-la perquè
                       ;                CLIC n'ha afectat una casella.
                  (RETOL-2 "Determinar" "quantes" "solucions" "normalitza-"
                           "des són" "compatibles" "pot trigar" "hores."
                           "Espereu...")
                  (PERFIL-V*V)
                  (COMPAT-PERFILS))
                (progn ; Hi ha FOTO2, amb 1 o més solucions ordinàries,
                       ;              però s'ha d'actualitzar perquè
                       ;              CLIC n'ha afectat una casella.
                  (RETOL-2 "Determinar" "quantes" "solucions" "compatibles"
                           "hi ha" "pot trigar" "una bona" "estona."
                           "Espereu...")
                  (ANORMALS ())))
            (command "DESHACER" "R" "UY" "ESPACIOM")))))
(if POST
    (progn
      (setq GR (strcat "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA."
                       "\nOmplint més caselles el fareu més fàcil"
                       " (<Intro> o < > per acabar)"))
      (if (> KK 1) ; Hi ha FOTO1 però cal seguir fins que només quedi
                   ;             1 solució normalitzada, o bé
                   ; hi ha FOTO2 però cal seguir fins que només quedi
                   ;             1 solució ordinària.
          (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                          (if KKK "" "normalitzades ")
                          "heu d'omplir més caselles:"))
          (if FOTO2
              (progn ; Hi ha FOTO2 i queda 1 solució ordinària:
                     ; si abans n'hi havia més cal actualitzar-la.
                (if KKK (setq FOTO2 (FOTO-V*V) KKK ())) ; La solució única no és
                ; normalitzada (no és ANORMALS qui troba KK = 1 sinó NUMCOMPAT).
                (prompt (strcat "\n.\n" GR)))
              (progn ; Hi ha FOTO1 i queda 1 solució normalitzada:
                     ; cal substituir-la per FOTO2 actual.
                (foreach E '(T ())
                  (prompt (strcat "\n.\nTeniu 1 solució normalitzada compatible"
                                  " amb V. E. NORMALITZADA."))
                  (if E (SEGUIR ()) (terpri)))
```

```
                    (command "ESPACIOP"
                             "DESHACER" "M"
                             "INSERT" "C" "0,0" "" "" "")
                    (RETOL-2 "Però pot" "haver-n'hi" "més, de no" "normalitza-"
                             "des però" "igualment" "compatibles." "Així que"
                             "espereu...")
                    (ANORMALS T)
                    (command "DESHACER" "R" "UY" "ESPACIOM")
                    (if (> KK 1)
                        (progn
                          (if (> KK 2)
                              (prompt (strcat "\n.\nHi ha " (itoa (1- KK))
                                              " solucions més, no normalitzades"
                                              " però també compatibles,"))
                              (prompt (strcat "\n.\nHi ha una solució més, no nor"
                                              "malitzada però també compatible,")))
                          (prompt "\naixí que heu d'omplir més caselles:"))
                        (prompt (strcat "\n.\nCom no n'hi ha cap de "
                                        "no normalitzada, " GR)))))))))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     **V**V**)
  (if (= 1-2 "1")
      (INI-LLL)
      (progn (NORMTEXT-9*9) (EXPLICACIO)))
  (setq PPM () **V**V** (INI-COORDS)
        NOPLUS (if (= 1-2 "1") **V**V**)
        **V*V** (if (= 1-2 "2") **V**V**)
        GR (if (= 1-2 "1") (strcat "Sobre la VARIANT ESCOLLIDA (dreta), feu clic "
                                   "en les caselles que hagi de veure\nel jugador"
                                   " (per anul•lar els últims clics, polseu la "
                                   "tecla \"<---Backspace\"):") "."))
  (prompt (strcat "\n.\n.\n" GR))
  (if (not (and (= ABC "A") (= 1-2 "1")))
      (command "ESPACIOM" "CVPORT" 2
               "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""))
  (command "CVPORT" 3 "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  (graphscr)
  (YIN-YANG)
  (while (not (or (and (= 1-2 "1") (COMPLET-9*9 **V**V**))
                  (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if (or (= 1-2 "2") (= (getvar "CVPORT") 3))
             (if (= (car GR) 5)
                 (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
                 (if (or (and (= 1-2 "1") PPM (equal GR '(2 8)))
                         (and (= (car GR) 3)
                              (> (getvar "CVPORT") 1)
                              (setq P (cadr GR) PX (car P) PY (cadr P))
                              (equal (list PX PY) '(4 4) 4.48)
                              (or (< (- PX (setq X (fix PX))) 0.48)
                                  (< (- (setq X (fix (1+ PX))) PX) 0.48))
                              (or (< (- PY (setq Y (fix PY))) 0.48)
                                  (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                              (or (= 1-2 "2") (listp (ELEMENT **V**V**)))
                              (PUNT)))
                     (if (= 1-2 "1") (FES-SUDOKU-1) (FES-SUDOKU-2))))))
  (if (= 1-2 "1")
      (progn
        (while (< (last (car PPM)) 2) (setq PPM (cdr PPM)))
        (setq SS ())
        (foreach E PPM (if (> (last E) 0) (setq SS (cons E SS))))
        (setq PPM SS SS (ssadd))
        (RASTRE))
      (command "ESPACIOP"))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" ""))
```

```
(defun C:SUDOKULUM (/ ANG TG L1A9 CONF ABC 1-2 G I J K L LL LLL M N N1 N2 N3 N4 P
                      PPM PX PY Q QX QY SS V X Y OK GR NOPLUS POST OSN FIL SRT SVT
                      ECO **9*9** **9**9**)
  (setq ANG -0.5   ; Un angle que giri el YIN-YANG a una velocitat proporcionada a
        TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9))                ; la del cursor.
  (PREPGRAF-9*9)
  (while (and (/= CONF "Si") (/= ABC "D"))
         (repeat 40 (terpri))
         (if (= CONF "No")
             (progn
               (if (> (getvar "CVPORT") 1) (command "ESPACIOP"))
               (command "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" "" "ESPACIOM"
                        "CVPORT" 2 "BORRA" "C" "0,0" "8,8" ""
                        "CVPORT" 3 "BORRA" "C" "0,0" "8,8" "")
               (***)
               (prompt "\n*******************************  NOVA SOLUCIÓ")
               (prompt "  *******************************")
               (***)
               (terpri)))
         (PREPTEXT-9*9)
         (if (= ABC "C")
             (setq **9*9** (list '(1 2 3 4 5 6 7 8 9) '(4 5 6 7 8 9 1 2 3)
                                 '(7 8 9 1 2 3 4 5 6)
                                 '(2 3 4 5 6 7 8 9 1) '(5 6 7 8 9 1 2 3 4)
                                 '(8 9 1 2 3 4 5 6 7)
                                 '(3 4 5 6 7 8 9 1 2) '(6 7 8 9 1 2 3 4 5)
                                 '(9 1 2 3 4 5 6 7 8)))
             (FES-SOLUCIO))
         (if (/= ABC "D")
             (progn
               (setq **9**9** **9*9**)
               (if (= ABC "C") (progn (command "ESPACIOM") (GRAF-9*9 ())))
               (command "ESPACIOP")
               (SEGUIR T)
               (RETOL-1 '(-0.75 -0.55) (strcat "SOLUCIÓ "
                                               (cond ((= ABC "A") "CREADA")
                                                     ((= ABC "B") "COPIADA")
                                                     (T "CANÒNICA"))))
               (RETOL-1 '(0.75 -0.55) "VARIANT ESCOLLIDA")
               (VARIACIONS)
               (CANON->VARIANT))))
  (if (/= ABC "D") (progn (TRIA-METODE) (FES-SUDOKU)))
  (prompt "\n.\n(Als efectes de revocació, l'execució de SUDÒKULUM ")
  (prompt "compta com una sola ordre.)")
  (command "OSMODE" OSN
           "FILLMODE" FIL
           "SORTENTS" SRT
           "SAVETIME" SVT
           "DESHACER" "F")
  (setvar "CMDECHO" ECO)
  (princ))
```

****

****

****

Actualització a l'última versió de FES-SOLUCIO del capítol *5- Etapa prèvia: crear una solució, III (roda el món i torna al Born...)*.

Dels canvis induïts a FES-SUDOKU pel pas de la versió precedent a la referenciada, els més importants tenen l'origen en la supressió parcial del dispositiu SUD-MIN per a l'activació de POST, decidint el moment en què la verificació de les normes primàries de compatibilitat de valors és insuficient i cal passar a la maquinària pesant RE-MEMBER. I parlem de supressió parcial no només en el sentit que SUD-MIN deixa d'intervenir en FES-SOLUCIO (on ja d'entrada fem POST = T) però no encara en FES-SUDOKU-1 (ni tampoc en FES-SUDOKU-2, que és l'únic lloc on la seva presència, que en aquest context serveix per posar en marxa la maquinària pesant PERFIL-V*V + + COMPAT-PERFILS, és plenament justificable), sinó perquè aquest canvi és purament

tàctic i conjuntural: havent comprovat que el grau de sofisticació assolit per RE-MEMBER permet d'utilitzar-lo de bon principi, sense causar l'alentiment tan temut inicialment, no necessitem més arguments. Hauríem de parlar d'inconseqüència, més que de parcialitat, perquè allò conseqüent hauria estat acceptar d'una vegada que SUD-MIN no hi fa res fixant el llindar d'actuació de RE-MEMBER, i adoptar el canvi estratègic consistent a eliminar SUD-MIN de tot arreu, tret de FES-SUDOKU-2. Però, perquè que el lector pugui copsar la contumàcia de l'autor (que, si no ha fotut el peu a totes les galledes trobades pel camí, de poques se n'ha lliurat), esperarem l'annex següent per recollir un factor circumstancial més i prescindir de SUD-MIN en l'àmbit de FES-SUDOKU-1, mantenint-nos fidels a l'equívoca decisió presa en el capítol *Condicions mínimes* on, malgrat l'encertat diagnòstic, havia deixat escrit:

> "Si el desencadenant és un criteri d'ocupació mínima, per sota de la qual no hi pot haver SUDOKU-PROBLEMA, tindrà sentit aplicar-lo a COMPAT-PERFILS, però fer-ho a RE-MEMBER sembla gratuït. Doncs bé: sí que ho és, d'arbitrari, però mentre en el camí no aparegui cap raó en contra, també l'usarem provisionalment com a llindar d'aplicació d'aquesta funció."

Començarem presentant el nou aspecte de les funcions ACTUALITZA, MASCARA i FES-SOLUCIO. La primera manté la necessitat d'actualitzar NOPLUS, i ho farà sota la mateixa condició (not (or POST PPLUS)) d'abans, tot i que ara només FES-SUDOKU-1 necessiti aquesta variable, perquè, funcionant amb POST = T de manera permanent, FES-SOLUCIO ja en queda exclosa. De MASCARA només es tracta de veure refoses la reordenació més acurada de 0-**9*9** i 0-LLL, pròpia de la versió de referència, amb l'adaptació als accessos des de FES-SUDOKU, mostrats en la versió precedent. I, de FES-SOLUCIO, podríem haver-nos limitat a dir que reprenia l'aspecte de la versió de referència, gràcies a la desaparició de la línia de codi (if (not POST) (SUD-MIN NOPLUS T)), si no hagués estat per dues foteses: el pas de la variable SS de local en aquesta funció a global en el programa, i que no hi figurava PPM.

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L LL PLUS PPLUS)
  (while (not PLUS)
         (setq X (car A-P) Y (cadr A-P)
               A-9*9 (ACT-9*9 A-N A-P A-9*9)
               A-LLL (ACT-LLL X Y A-LLL))
         (if (not (or POST PPLUS)) (setq NOPLUS (ACT-9*9 A-N A-P NOPLUS)))
         (ACTUALITZA-PLUS)
         (setq PLUS (not PLUS) PPLUS T))
  (list A-9*9 A-LLL))

(defun MASCARA (L / INI OK PROU Ñ JJ 9**9 0-**9*9** 0-LLL 0-L)
  (if (not 1-2) (setq LL () M 0 SS (ssadd)))
  (if POST
      (progn
        (setq N (N-3*3 (if 1-2 **V**V** **9*9**)) Ñ (TRANSPOSAR N)
              N (1+2+3 N) Ñ (1+2+3 Ñ))
        (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
            (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
        (if (and (>= (car N) (cadr N)) (>= (cadr N) (caddr N)))
            (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL)
            (progn
              (setq JJ (if (and (<= (car N) (cadr N)) (<= (cadr N) (caddr N)))
                           '(2 1 0)
                           (if (>= (car N) (caddr N))
                               (if (> (cadr N) (caddr N)) '(1 0 2) '(0 2 1))
                               (if (> (cadr N) (caddr N)) '(1 2 0) '(2 0 1))))
                    0-LLL (F=3 (if 1-2 **V**V** **9*9**)) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**) 0-LLL (F=3 LLL)
                    P (list X (+ (rem Y 3)
                                 (* 3 (- 3 (length (member (/ Y 3) JJ)))))))))
        (if (< (car Ñ) (caddr Ñ))
            (progn
              (setq K ())
```

```
            (foreach EE (reverse 0-**9*9**)
              (setq 0-L ())
              (foreach E EE
                (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                0-L)))
              (setq K (cons 0-L K)))
            (setq 0-**9*9** K K ())
            (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
            (setq 0-LLL K P (list (- 8 (car P)) (cadr P)))))))
  (if (and POST 1-2)
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")))
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
    (if (and (member N L)
             (not (or PROU (= N GR)))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if 1-2
              (setq PROU T)
              (progn
                (if POST (TREU-RETOL))
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
                (command "DESIGNA" (ssadd (entlast) SS) ""))))
          (if (and POST (not 1-2)) (TREU-RETOL))))
  (if POST
      (progn
        (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))

(defun FES-SOLUCIO (/ CURSOR P-CURS C->F)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq PPM () **9*9** (INI-COORDS)
              P-CURS '(-1.1919 0.4419) POST T)
        (INI-LLL)
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP" "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (TECLA-P ())
               (command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
               (setq PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
                     LLL (ACTUALITZA **9*9** N P LLL T)
                     **9*9** (car LLL) LLL (cadr LLL)))
        (while (= (last (car PPM)) 1) (setq PPM (cdr PPM)))
        (setq POST () SS (ssadd))
        (prompt "\n.\nPodeu seguir, per construir el SUDOKU-PROBLEMA (el joc) ")
        (prompt "a partir d'aquesta\nSUDOKU-SOLUCIÓ, o acabar i quedar-vos amb ")
```

```
        (prompt "el que heu definit sobre la marxa.")
        (initget "Si No")
        (setq CONF (getkword "\nVoleu seguir (S/N)? <S>: ")
              CONF (if CONF CONF "Si"))
        (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
                 "CAMBPROP" "P" "" "C" "VAR-1" "")
        (if (= CONF "Si") (command "ESPACIOP") (RASTRE)))
      (progn
        (prompt "\n.\n.\nCLIC sobre el nombre que va a la casella marcada: ")
        (setq K 9)
        (repeat 9
          (setq L () J -1 K (1- K))
          (repeat 9
            (setq J (1+ J))
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.110465,0" "" "ESPACIOM"
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N) "ESPACIOP")
            (setq L (append L (list N))))
          (command "DESPLAZA" CURSOR "" "-0.9942,-0.1105" "")
          (setq **9*9** (cons L **9*9**)))))
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))
```

Si ho deixem així, en el nou camí a què fa referència el títol del nostre capítol (mètode 1) subsisteix una deficiència relativa a l'ús de la tecla "<---Backspace": Si fem ús d'aquesta tecla per revocar l'última casella activada manualment però es dóna la circumstància que després d'aquest emplenament se n'havia produït algun d'automàtic (algun forat tapat per ACTUALITZA-PLUS), seria lògic que la revocació s'endugués de retruc els acompanyants lligats indissolublement al primer, però tal i com ho hem deixat la cosa no anirà pas així, sinó que haurem de revocar un a un tots els forats tapats a conseqüència de l'emplenament desencadenant abans que li arribi el torn a aquest. I no es tracta simplement d'una qüestió d'incomoditat, o de la possibilitat que l'usuari es limiti a polsar un sol cop la tecla esmentada, deixant només activada la casella desencadenant (amb, eventualment, alguns forats tapats, tret de l'últim): si s'hagués previst correctament no hi hauria conflicte, perquè en la següent activació (amb la consegüent intervenció d'ACTUALITZA-PLUS) el forat destapat es tornaria a tapar; però l'afany d'adaptar massa expeditivament FES-SUDOKU-1 a les revocacions, sense atendre les singularitats dels emplenaments automàtics, ens ha traït. Per poder seguir en detall el que passa, sense que el lector hagi de fullejar amunt i avall, portarem aquí la resta de funcions en joc:

```
(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (if 1-2
          (progn
            (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
                  PPM (cons (list P1 P2 0) PPM)
                  LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                                LLL/LLL))
            (command "COPIA" "C" P1 P2 "" "0,0" ""
                     "CAMBPROP" "LT" "" "C" "VAR-1" ""))
          (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N)))))

(defun COMMUTA (LL)
  (subst (subst (if V (list X Y) N) (if V N (list X Y)) (nth Y LL))
         (nth Y LL) LL))

(defun V+- (C)
  (if V
      (command "BORRA" (ssname SS 0) "")
      (command "COPIA" SS "" "0,0" ""
               "CAMBPROP" "LT" "" "C" C "")))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if (and (= 1-2 "2") Q) (setq P Q X (car P) Y (cadr P) PX QX PY QY))
```

```
        (setq N (ELEMENT **9**9**)
              V (atom (ELEMENT **V**V**))
              **V**V** (if (and (= 1-2 "1") (not V)) **V**V** (COMMUTA **V**V**))
              SS (ssget "_C" PX PY))
        (if (and (= 1-2 "2") (not Q)) (setq Q P QX PX QY PY))
        (V+- "VAR-1") ...) ...))

(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM) PY (cadar PPM)
            X (fix (/ (+ (car PX) (car PY)) 2))
            Y (fix (/ (+ (cadr PX) (cadr PY)) 2)) P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (and (not POST) (> (last (car PPM)) 0))
            (setq NOPLUS (COMMUTA NOPLUS)))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (setq M 1 GR N)
        (if (not LLL) (INI-LLL))
        (if (not POST) (SUD-MIN NOPLUS T))
        (MASCARA (ELEMENT LLL))
        (setq N GR PPM (cons (list PX PY M) PPM)
              LLL (ACTUALITZA **V**V** N P LLL T)
              **V**V** (car LLL) LLL (cadr LLL)
              LLL/LLL (cons LLL LLL/LLL)))))
```

En activar una casella (diem-ne P_a) que provoca l'activació automàtica d'una altra (diem-ne P_b), les coses a FES-SUDOKU-1 (V = nil) es produeixen en l'ordre següent:

- A CLIC, **V**V** no experimenta cap canvi (P_a segueix buida, representada per les coordenades) i, per acció de V+-, el contingut de P_a passa de gris (capa VAR-0) a blanc (es crea una còpia a la capa VAR-1).
- P_a s'incorpora a la llista PPM, amb una marca 2 o 1 (segons que MASCARA hagi revelat que té o no més candidats vàlids que el valor prefixat).
- A PLUS-N->P < ACTUALITZA-PLUS < ACTUALITZA, P_b s'incorpora a la llista PPM, amb una marca 0, i una versió de LLL que recull la presència de P_b s'incorpora a la llista LLL/LLL.
- **V**V** i LLL s'actualitzen recollint la presència de P_a i P_b, i aquesta versió de LLL s'incorpora a la llista LLL/LLL.

Si immediatament després d'això es polsa la tecla "<---Backspace" per desactivar la casella P_b, les coses a FES-SUDOKU-1 (V = T) es produeixen en l'ordre següent:

- A CLIC, COMMUTA provoca que en **V**V** P_b es buidi (passi a estar representada per les coordenades) i, per acció de V+-, el contingut passi de blanc (s'esborra la còpia de la capa VAR-1) a gris (capa VAR-0).
- S'esborra l'últim element (el corresponent a P_b) incorporat a la llista PPM, i el corresponent a P_a restarà com a últim element incorporat. S'esborra l'últim element (la pseudomatriu LLL que recollia la presència de P_a i P_b) incorporat a la llista LLL/LLL, i el corresponent a la LLL que només recollia la presència de P_b restarà com a últim element incorporat.
- LLL s'actualitza identificant-se amb l'últim element esmentat, amb la qual cosa tindrem una pseudomatrius 9×9 **V**V** amb la casella P_a ocupada i la P_b buida, d'acord amb l'aspecte visual de l'escaquer VARIANT ESCOLLIDA (dreta), però una pseudomatrius 9×9 LLL corresponent a una P_a buida i a una P_b ocupada.

A partir d'aquesta incoherència entre **V**V** i LLL ja es comprèn que pot passar de tot. És clar que amb qualsevol artifici podríem aconseguir que els elements que incorpora PLUS-N->P a LLL/LLL ho fessin després que aquesta llista incorporés una LLL actualitzada només amb la casella desencadenant P_a, però ja que consideràvem més convenient revocar en bloc P_a i la seqüència de forats P_b tapats, la falta de correspondència entre **V**V** i LLL a nivell de casella no s'apreciarà, perquè només comptaran la situació anterior i posterior a la revocació del bloc, en què **V**V** i LLL tornen a estar sincronitzats. N'hi haurà prou a refer FES-SUDOKU-1:

```
(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM) PY (cadar PPM)
            X (fix (/ (+ (car PX) (car PY)) 2))
            Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
            P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (and (not POST) (> (last (car PPM)) 0))
            (setq NOPLUS (COMMUTA NOPLUS)))
        (if (= (last (car PPM)) 0)
            (while (< (last (car PPM)) 1)
                   (setq PPM (cdr PPM)
                         LLL/LLL (cdr LLL/LLL)
                         LLL (car LLL/LLL)
                         PX (caar PPM) PY (cadar PPM)
                         X (fix (/ (+ (car PX) (car PY)) 2))
                         Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
                         N (ELEMENT **9**9**)
                         **V**V** (COMMUTA **V**V**)
                         SS (ssget "_C" PX PY))
                   (V+- "VAR-1")))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (setq M 1 GR N)
        (if (not LLL) (INI-LLL))
        (if (not POST) (SUD-MIN NOPLUS T))
        (MASCARA (ELEMENT LLL))
        (setq N GR PPM (cons (list PX PY M) PPM)
              LLL (ACTUALITZA **V**V** N P LLL T) **V**V** (car LLL)
              LLL (cadr LLL) LLL/LLL (cons LLL LLL/LLL)))))
```

Algú pot pensar que, ja que les seqüències ininterrompudes de forats tapats s'han de revocar d'una tirada, conjuntament amb la casella desencadenant, el més pràctic seria no incorporar-los a la llista PPM, prescindint conseqüentment de la marca 0. En aquesta línia, podríem incórrer en el simplisme de creure que amb bandejar de PLUS-N->P tota referència a PPM i LLL/LLL; amb incloure el parell d'instruccions `(if (= 1-2 "1") (command "DESHACER" "R"))` i `(if (= 1-2 "1") (command "DESHACER" "M"))` a la funció V+-, en el primer i últim lloc, respectivament, i amb situar-les també en FES-SUDOKU, immediatament abans i després del motor d'iteracions while (la primera després de (YIN-YANG) i la segona integrada en l'expressió condicional posterior, que podríem simplificar, reduïda a

```
(if (= 1-2 "1")
    (progn
      (while (< (last (car PPM)) 2) (setq PPM (cdr PPM)))
      (setq SS (ssadd))
      (command "DESHACER" "R")
      (RASTRE))
    (command "ESPACIOP"))
```

, en haver desaparegut les marques 0 de la llista PPM). Però amb aquests únics retocs no n'hi hauria prou: tot aniria bé fins que se'ns acudís de revocar una activació amb cua (tapament d'un o més forats), perquè el dispositiu donaria una visualització correcta però de portes endins només buidaria les caselles activades manualment: les d'activació automàtica haurien esborrat el seu rastre de LLL però encara ocuparien **V**V** perquè (COMMUTA **V**V**), des de CLIC, s'hauria limitat a desocupar la casella desencadenant. Recuperar el control de la situació passa necessàriament per enregistrar en una llista els emplenaments automàtics esdevinguts a PLUS-N->P, aprofitant el tàndem PPM / LLL/LLL o creant una estructura diferenciada (una llista constituïda per subllistes representant les seqüències ininterrompudes de forats tapats) que FES-SUDOKU-1 depuraria, cada cop que les afectés una revocació, mitjançant un dispositiu no molt més senzill que el de la versió que hem proposat unes línies amunt. A la pàgina 419 ja havíem pronosticat que "l'alternativa a la inclusió dels forats tapats en PPM passaria inexorablement per la creació d'una llista específica, solució poc recomanable perquè encara ens ho complicaria més tot". A més, aquesta alternativa portaria un efecte col·lateral, no massa greu però sí desagradable per a l'usuari: cada vegada que una pulsació "<---Backspace" destapés forats, el símbol del YIN-YANG, la lenta rotació del qual ve a ser la tranquilitzadora confirmació visual d'estar actuant

correctament, mentre el cursor es mou cap a la propera casella (a diferència del mètode 2, que admet fer clics en la finestra de l'esquerra o en la de la dreta, seguint el mètode 1 només resta operativa la segona, circumstància que no trigaran a copsar els despistats que s'hagin passat a la finestra de l'esquerra, en veure que el símbol no es mou amb el cursor, fins i tot abans de fer clic i adonar-se que això no provoca cap mena d'alteració), experimentaria un inesperat retrocés. Tot i haver-ho tractat en el capítol *Condicions mínimes*, aprofitarem per recordar que els forats tapats no es compten, als efectes del límits establerts en SUD-MIN.

Ja hem assenyalat en aquest mateix capítol, abans de mostrar la versió integra del codi, que amb el mètode 2 no hi ha emplenament automàtic de forats ni tampoc cal desactivar les caselles més recents abans de revocar-ne una de qualsevol, raó per la qual aquesta problemàtica no l'afecta.

****
****
****

Actualització a l'última versió de FES-SOLUCIO del capítol *6- Etapa prèvia: crear una solució, IV (... o a Camprodon)*.

La necessitat d'identificar quan abans millor configuracions de tipus AAA-AAA o AAA-BBB, per poder explicitar tercets implícits en LLL (A-LLL o R-LLL), és causa de la major part de modificacions en l'àmbit de FES-SUDOKU-1, degudes a dos fets:

- **Tot i que hauríem pogut introduir una excepció en SUD-MIN (amb PREVI = T) perquè POST s'activés amb 6 caselles emplenades, 3 en un requadre 3×3 i 3 més en un altre d'alineat horitzontalment o vertical (només amb això pot quedar definida una configuració de les esmentades, pel que vèiem en el capítol de referència), o, sense filar tan prim, limitar-nos a rebaixar de 17 a 12 el nombre total de caselles ocupades (amb 12 pot haver-hi configuracions AAA-AAA dobles o AAA-BBB per damunt del requadre 9×3 inferior, que també causen demores inacceptables), a FES-SUDOKU-1 sembla més pràctic posar en marxa la maquinària pesant RE-MEMBER de bon començament, activant POST d'entrada i ometent-hi tota referència a SUD-MIN. De fet, aprofitem aquesta avinentesa (i la progressiva eficiència de RE-MEMBER, que minimitza els efectes nocius d'una implantació precoç) per desempallegar-nos d'un instrument que, tret de l'àmbit de FES-SUDOKU-2, utilitzàvem impròpiament.**
- **Quan FES-SUDOKU-1 activi la casella que faltava per determinar una configuració AAA-AAA o AAA-BBB, caldrà depurar de valors innecessaris les llistes que formen LLL (el que anomenàvem explicitar tercets implícits), de cara a les activacions següents. Això significa que a MASCARA no n'hi haurà prou a veure si hi ha o no més candidats, a banda del predeterminat (el que anem a activar), propòsit que ens permetia saltar-nos-el si no hi havia cap exigència relacionada amb aquest últim valor, en l'enquesta feta per aquesta funció. Ara serà just el contrari: tot i saber que el veredicte de RE-MEMBER sobre el candidat N = GR és favorable, aquest haurà de passar un altre cop per l'adreçador per depurar LLL, si s'escau, i així evitar que en alguna activació posterior el veredicte sobre un candidat alternatiu (un valor diferent del predeterminat GR) ens obligui a esperar massa. (Quan parlem de depurar LLL -A-LLL o R-LLL- de cara a posteriors activacions i de fer-ho únicament amb el candidat N = GR, volem referir-nos a guardar aquesta informació a TN/R-L per tal que ACTUALITZA -ja amb REAL = T- pugui fer-ne ús més endavant, perquè en realitat tots els candidats que accedeixin RE-MEMBER hauran de passar per TERCETS i, si s'escau, per FILS/COLS, per poder expurgar R-LLL: fins i tot quan N ≠ GR, caldrà completar l'exploració per arribar a un veredicte sobre el candidat, i la depuració pot ser necessària per no allargar-la massa.)**

Teòricament, la mateixa argumentació (que 6 caselles plenes poden donar lloc a una configuració AAA-AAA o AAA-BBB) també forçaria a revisar l'àmbit de FES-SUDOKU-2: no ja el procediment inicial (no importa que, en activar SUD-MIN POST, ens trobem de cop i volta amb solucions normalitzades compatibles que acullin aquest tipus de configuracions, perquè juguem amb solucions acabades) sinó quan, havent arribat a un sudoku a mig fer compatible amb una única solució normalitzada (i, en general, amb més solucions no normalitzades) i havent tret FOTO2 almenys per primer cop, a l'usuari se l'acut desactivar una casella que sortia en aquesta foto; llavors sí que RE+MEMBER < ANORMALS haurà d'escometre una exploració encara més feixuga que RE-MEMBER < MASCARA (no en té prou a mirar si hi ha alguna solució compatible amb cadascun dels valors candidats a la casella actual, sinó que haurà de trobar totes les solucions compatibles amb l'emplenament que queda en buidar aquesta casella), tot i que el grau d'ocupació serà força més elevat que el corresponent al moment d'activació de POST, tret que l'usuari s'entesti a multiplicar les desactivacions.

En aquest supòsit (havent-hi mots escaients com "supòsit", "circumstància", "cas" o "hipòtesi", resulta irritant la moda de recórrer a "escenari" a tort i a dret), sí que podríem considerar que ANORMALS incorporés un dispositiu similar al tàndem TERCETS + FILS/COLS, després d'haver tret la FOTO2 i reconstruït LLL, i just abans d'entrar en el terç final d'aquesta funció (accés a RE+MEMBER, amb FOTO1 o FOTO2), amb el propòsit d'expurgar LLL, si s'escau, explicitant-ne els tercets implícits. Però les experiències realitzades abans de posar-nos-hi ens han revelat que no és imprescindible: de fet, els temps d'espera per conèixer el nombre de solucions no normalitzades compatibles amb les caselles activades eren tan raonables en el cas de conflicte amb configuracions AAA-AAA o AAA-BBB com en el cas de no haver-n'hi; l'alta ocupació evitava que el conflicte donés lloc a una exploració massa llarga.

Ara a MASCARA el comptador M s'incrementarà en una unitat en processar el candidat predefinit N = GR, raó per la qual haurem de posar-lo a 0, no a 1: treurem aquesta inicialització de FES-SUDOKU-1 i aprofitarem la de MASCARA, que s'estendrà a tots els altres accessos a la funció; així doncs, quedaria fora del condicionament a (not 1-2), però no perdem temps mostrant-ne l'aspecte, perquè no trigarem a veure com LL i SS també queden fora, dintre d'una operació més àmplia de simplificació. Seguint amb els canvis en aquesta funció, sembla que n'hi hauria prou a substituir les expressions subratllades en el fragment

```
  ..................................................................
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
    (if (and (member N L)
             (not (or PROU (= N GR)))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if 1-2
              (setq PROU T)
              (progn ...)) ...) ........................................
```

per aquestes altres, també subratllades,

```
  ..................................................................
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
    (if (and (member N L)
             (not PROU)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if 1-2
              (if (and (> M 1) (>= N GR)) (setq M 2 PROU T))
              (progn ...)) ...) ........................................
```

Tot i que hi ha una petita pega: amb més d'un candidat N < GR que hagi passat la condició (member N L), pot ser que un d'ells hagi rebut un veredicte favorable i tanmateix, abans d'arribar al valor predefinit N = GR, la resta sigui processada innecessàriament per RE-MEMBER. Podríem evitar-li aquest tràmit inútil fent això:

```
  ..................................................................
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
    (if (and (member N L)
             (not PROU)
             (or (not 1-2) (= N GR) (and (< M 2) (not (and (= M 1) (< N GR)))))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if 1-2
              (if (= M 2) (setq PROU T))
              (progn ...)) ...) ........................................
```

Però, havent-ho deixat així, PROU es del tot superflu. En prescindirem, i quedarà

```
  ..................................................................
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
```

```
    (if (and (member N L)
             (or (not 1-2) (= N GR) (and (< M 2) (not (and (= M 1) (< N GR)))))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-2)
              (progn ...)) ...) ........................................
o bé, substituint l'expressió subratllada per una d'equivalent i més simple,
  ..........................................................................
  (foreach N L1A9
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "CLIC sobre" "el nombre..."))
    (if (and (member N L)
             (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-2)
              (progn ...)) ...) ........................................

Així que MASCARA adoptaria la forma següent:

(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP*)
  (setq M 0)
  (if (not 1-2) (setq LL () SS (ssadd)))
  (if POST
      (progn
        (setq N (N-3*3 (if 1-2 **V**V** **9*9**)) Ñ (TRANSPOSAR N)
              N (1+2+3 N) Ñ (1+2+3 Ñ))
        (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
            (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
        (PERMUT-JJ N)
        (setq JJ2 JJ)
        (if JJ
            (progn
              (setq 0-LLL () I -1)
              (foreach EE (F=3 (if 1-2 **V**V** **9*9**) T)
                (setq 0-L () I (1+ I))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) I)) 0-L)))
                (setq 0-LLL (cons (reverse 0-L) 0-LLL)))
              (setq 0-**9*9** (reverse 0-LLL) 0-LLL (F=3 LLL T)
                    P (list X (+ (rem Y 3)
                                 (* 3 (- 3 (length (member (/ Y 3) JJ))))))))
            (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL))
        (setq *P () P* () J -1)
        (repeat 3
          (setq 0-9 () 0-L () N ())
          (repeat 3
            (setq J (1+ J) K 0)
            (foreach E (nth J 0-**9*9**)
              (if (or (atom E) (equal E P)) (setq K (1+ K))))
            (setq N (cons K N) 0-9 (cons (nth J 0-**9*9**) 0-9)
                  0-L (cons (nth J 0-LLL) 0-L)))
          (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
          (PERMUT-JJ N)
          (setq JJ1 (cons JJ JJ1))
          (if JJ
              (progn
                (setq *PP* () I (- J 3))
                (if (and (> (cadr P) I) (<= (cadr P) J))
                    (setq P (list X (1+ (- J (length (member (rem (cadr P) 3)
                                                             JJ)))))))
                (foreach EE (F=3 0-9 ())
                  (setq 0-9 () I (1+ I))
                  (foreach E EE
                    (setq 0-9 (cons (if (atom E) E (list (car E) I)) 0-9)))
```

```
                (setq *PP* (cons (reverse 0-9) *PP*)))
              (setq *P (append *P (reverse *PP*)) P* (append P* (F=3 0-L ())))
            (setq *P (append *P 0-9) P* (append P* 0-L))))
        (setq JJ1 (if (equal JJ1 '(() () ())) () (reverse JJ1))
              0-**9*9** *P 0-LLL P*)
        (if (< (car Ñ) (caddr Ñ))
            (progn
              (setq INV-F T K ())
              (foreach EE (reverse 0-**9*9**)
                (setq 0-L ())
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                  0-L)))
                (setq K (cons 0-L K)))
              (setq 0-**9*9** K K ())
              (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
              (setq 0-LLL K P (list (- 8 X) (cadr P)))))))
  (if (and POST 1-2)
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")))
  (foreach N L1A9
    (if (and POST (not 1-2))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                 "el nombre...\""))
    (if (and (member N L)
             (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-2)
              (progn
                (if POST (TREU-RETOL))
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
                (command "DESIGNA" (ssadd (entlast) SS) ""))))
        (if (and POST (not 1-2)) (TREU-RETOL))))
  (if POST
      (progn
        (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))
```

Amb la supressió dels accessos a SUD-MIN tant des de FES-SOLUCIO (decidida en el subcapítol *Etapa prèvia: crear una solució, III (roda el món i torna al Born...)*) com des de FES-SUDOKU-1, ¿no seria més lògic eliminar POST com a variable global (en el sentit de ser local a C:SUDOKULUM) i confinar-la a la funció FES-SUDOKU-2?: així ja no caldria activar-la a l'inici de les dues primeres funcions esmentades i, sobretot, evitaríem que la funció MASCARA, reclamada des d'una i altra (i des d'una funció instrumental de C:SUDOKULUM com 2-9V < CANON->VARIANT), fos una via d'allò més accidentada precisament per l'omnipresència de POST. En teoria sí, però si recordem que les crides (MASCARA L1A9), fetes des de TECLAT < FES-SOLUCIO o des de TECLAT < 2-9V, i (MASCARA MASC) feta directament des d'aquesta última funció), totes en condicions POST = nil, eren el recurs utilitzat per a la presentació del caseller central 9×9 ple, no queda tan clar. Ningú no nega que podríem incorporar a MASCARA un argument binari, diguem-ne POTI, de manera que (MASCARA L1A9) passés a ser (MASCARA L1A9 ()) i (MASCARA (ELEMENT LLL)) a ser (MASCARA (ELEMENT LLL) T), però si anéssim a concretar l'estalvi realitzat ens trobaríem que:

- La sentència (if (not 1-2) ...) i les dues (if 1-2 ...) es mantenen.
- Les tres sentències (if POST ...) passen a ser (if POTI ...).
- La sentència (and POST 1-2) i les dues (and POST (not 1-2)) es converteixen en (and POTI 1-2) i (and POTI (not 1-2)), respectivament.
- L'expressió (not POST) passa a ser (not POTI).

Com veieu, seria un simple canvi de nom. Però segregar la part central de MASCARA (en realitat l'últim terç, tret de les 5 línies finals), convertint-la en aquesta funció MASCA,

```
  (defun MASCA (L POTI)
    (setq LL () M 0 SS (ssadd))
    (foreach N L1A9
      (if (and POTI (not 1-2))
          (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                   "el nombre...\""))
      (if (and (member N L)
               (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
               (or (not POTI) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
          (progn
            (setq LL (cons N LL) M (1+ M))
            (if (not 1-2)
                (progn
                  (if POTI (TREU-RETOL))
                  (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                                 .......................
                                 (   T     '(0.15 -0.15)))
                           (itoa (if MASC (nth (1- N) MASC) N)))
                  (command "DESIGNA" (ssadd (entlast) SS) "")))))
          (if (and POTI (not 1-2)) (TREU-RETOL))))
```

invocada directament des de TECLAT i 2-9V ((MASCA L1A9 ()) i (MASCA MASC ()), en lloc de (MASCARA L1A9 ()) i (MASCARA MASC ())), i des de MASCARA (on ja només s'hi accedia com a (MASCARA (ELEMENT LLL) T), (MASCA L T)), no comportarà gran estalvi, perquè d'aquesta última funció únicament les parts inicial i final es deslliuraran dels condicionaments a POTI, però almenys aconseguirem que el seu codi explícit no sigui tan llarg. Ara bé, aclarit que el segon argument de MASCA, tingui el nom que tingui, no té funcionalment res a veure amb la variable POST relegada a l'àmbit de FES-SUDOKU-2, deixarem de potinejar amb símbols estrafolaris i, per simplificar el poti-poti de noms, en substitució de POTI recuperarem POST. I, més enllà d'aquesta qüestió formal, observeu que no només el tram central de codi ha estat transferit a MASCA, sinó també la primera línia (if (not 1-2) (setq LL () M 0 SS (ssadd))), tot i que tan desprovista de condicionaments com (setq LL (cons N LL) M (1+ M)), 9 línies avall. De fet, només necessitem LL quan accedim a MASCA < MASCARA des de FES-SOLUCIÓ; SS, a més d'aquest cas, també és necessària a MASCA < TECLA, tant a TECLA < FES-SOLUCIO com a TECLA < 2-9V, i en l'accés directe des d'aquesta última; pel que fa a M, per acabar, dues pàgines enrera hem indicat que la inicialització es traslladava des de FES-SUDOKU-1 a MASCARA (ara a MASCA < MASCARA), és a dir que és l'única que necessitem tant amb 1-2 = nil (des de FES-SOLUCIO) com amb 1-2 = T (des de FES-SUDOKU-1 < FES-SUDOKU). Però deixant-les lliures les tres no crearem conflictes (només assignacions supèrflues) i evitarem farcir MASCA amb condicions.

Abans de passar a l'entorn de FES-SUDOKU-2, recollirem el codi que s'ha modificat per atendre els requeriments de FES-SUDOKU-1, d'acord amb allò que hem exposat en les últimes 5 pàgines. Tot i haver-nos prodigat especialment amb la funció MASCARA (i la nova subordinada MASCA), els canvis afecten igualment PLUS-N->P, ACTUALITZA, RE-MEMBER, TRIA-METODE i FES-SUDOKU-1, ni que sigui mínimament: així, a PLUS-N->P i ACTUALITZA cal afegir-hi les instruccions necessàries per visualitzar els forats tapats (com que són les mateixes, hem optat per passar-les a la nova funció P1P2), atenent els casos (not (cdr (assoc A-N TN/R-L))) i (cdr (assoc A-N TN/R-L)) de la segona, quan REAL = T; en accedir a RE-MEMBER i detectar configuracions AAA-AAA o AAA-BBB que donen lloc a depuracions de R-LLL, només ens molestem a transferir-les a TN/R-L (per ser recuperades en ACTUALITZA) amb el candidat predeterminat N = GR; a TRIA-METODE eliminem del text explicatiu el parèntesi "(a partir d'un cert punt, l'usuari haurà d'esperar autorització per activar una nova casella, però els temps d'espera cada vegada seran més breus)", perquè ja no existeix el punt de ruptura que marcava SUD-MIN, sinó que els temps d'espera (menors que a la versió anterior) hi són des del principi; de FES-SUDOKU-1 hem suprimit la inicialització (setq M 1) i les dues línies que fan referència a la variable NOPLUS i a la funció SUD-MIN.

```
(defun P1P2 (/ P1 P2)
  (if 1-2
      (progn
        (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
              PPM (cons (list P1 P2 0) PPM)
              LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                            LLL/LLL))
        (command "COPIA" "C" P1 P2 "" "0,0" ""
                 "CAMBPROP" "LT" "" "C" "VAR-1" ""))
      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa A-N))))

(defun PLUS-N->P (M)
  (setq A-N Ñ A-P M PLUS T)
  (if REAL
      (P1P2)
      (if INI (setq REPLUS (ACT-9*9 Ñ M REPLUS)))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y L PLUS)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
              A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-P (list J K) A-N E
                        A-9*9 (ACT-9*9 A-N A-P A-9*9))
                  (P1P2))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P)
                   A-9*9 (ACT-9*9 A-N A-P A-9*9)
                   A-LLL (ACT-LLL X Y A-LLL))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA 2R R-N R-P)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)
                           TN/R-L (if (or (not 1-2) (= N GR))
                                      (subst (list N (REORD R-LLL) (REORD REPLUS))
                                             (cons N TT) TN/R-L)
                                      (if (and (> N GR) (assoc GR TN/R-L))
                                          (list (assoc GR TN/R-L))))))))))
  (if (not (or INI OK))
      (if (not (setq OK (COMPLET-9*9 R-9*9)))
          (progn
            (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
            (foreach E L1A9
              (if (and (not OK) (member E R-N))
                  (progn
                    (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                    (if (not FORA)
                        (if (equal (cadr 2R) NIL*NIL)
                            (setq OK T)
                            (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)
```

```
(defun MASCA (L POST)
  (setq LL () M 0 SS (ssadd))
  (foreach N L1A9
    (if (and POST (not 1-2))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                 "el nombre...\""))
    (if (and (member N L)
             (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-2)
              (progn
                (if POST (TREU-RETOL))
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
                (command "DESIGNA" (ssadd (entlast) SS) ""))))
        (if (and POST (not 1-2)) (TREU-RETOL)))))

(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP*)
  (setq N (N-3*3 (if 1-2 **V**V** **9*9**)) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ))
  (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
      (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
  (PERMUT-JJ N)
  (setq JJ2 JJ)
  (if JJ
      (progn
        (setq 0-LLL () I -1)
        (foreach EE (F=3 (if 1-2 **V**V** **9*9**) T)
          (setq 0-L () I (1+ I))
          (foreach E EE (setq 0-L (cons (if (atom E) E (list (car E) I)) 0-L)))
          (setq 0-LLL (cons (reverse 0-L) 0-LLL)))
        (setq 0-**9*9** (reverse 0-LLL) 0-LLL (F=3 LLL T)
              P (list X (+ (rem Y 3) (* 3 (- 3 (length (member (/ Y 3) JJ)))))))
      (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL))
  (setq *P () P* () J -1)
  (repeat 3
    (setq 0-9 () 0-L () N ())
    (repeat 3
      (setq J (1+ J) K 0)
      (foreach E (nth J 0-**9*9**) (if (or (atom E) (equal E P)) (setq K (1+ K))))
      (setq N (cons K N)
            0-9 (cons (nth J 0-**9*9**) 0-9) 0-L (cons (nth J 0-LLL) 0-L)))
    (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
    (PERMUT-JJ N)
    (setq JJ1 (cons JJ JJ1))
    (if JJ
        (progn
          (setq *PP* () I (- J 3))
          (if (and (> (cadr P) I) (<= (cadr P) J))
              (setq P (list X (1+ (- J (length (member (rem (cadr P) 3) JJ)))))))
          (foreach EE (F=3 0-9 ())
            (setq 0-9 () I (1+ I))
            (foreach E EE (setq 0-9 (cons (if (atom E) E (list (car E) I)) 0-9)))
            (setq *PP* (cons (reverse 0-9) *PP*)))
          (setq *P (append *P (reverse *PP*)) P* (append P* (F=3 0-L ()))))
        (setq *P (append *P 0-9) P* (append P* 0-L))))
  (setq JJ1 (if (equal JJ1 '(() () ())) () (reverse JJ1)) 0-**9*9** *P 0-LLL P*)
```

```
  (if (< (car Ñ) (caddr Ñ))
      (progn
        (setq INV-F T K ())
        (foreach EE (reverse 0-**9*9**)
          (setq 0-L ())
          (foreach E EE
            (setq 0-L (cons (if (atom E)
                  E (list (- 8 (car E)) (cadr E))) 0-L)))
          (setq K (cons 0-L K)))
        (setq 0-**9*9** K K ())
        (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
        (setq 0-LLL K P (list (- 8 X) (cadr P)))))
  (if 1-2
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")))
  (MASCA L T)
  (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
  (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
  (setq P (list X Y)))

(defun TRIA-METODE ()
  (textscr)
  (prompt "\n\n\n\n\nDisposeu de dos mètodes per obtenir un SUDOKU-PROBLEMA ")
  (prompt "d'una SUDOKU-SOLUCIÓ\n(aquesta VARIANT ESCOLLIDA) reactivant caselles")
  (prompt " (deixant-les visibles):")
  (prompt "\n\n1) Anar-ho fent fins haver activat tota la solució, moment en què")
  (prompt " el programa\n   remuntarà el procés per només deixar visibles les ")
  (prompt "caselles imprescindibles.")
  (prompt "\n\n2) Anar-ho fent fins que el programa ens informi que només hi ha ")
  (prompt "una solució\n   compatible amb les caselles activades ")
  (prompt "(SUDOKU-PROBLEMA): la SUDOKU-SOLUCIÓ\n   (entremig del procés es ")
  (prompt "produiran dues pauses, que poden ser ben llargues,\n   però tret ")
  (prompt "d'això l'activació de cada casella -revisable- serà immediata).")
  (initget "1 2")
  (setq 1-2 (getkword "\n\nTrieu l'opció 1 o 2 <2>: ") 1-2 (if 1-2 1-2 "2")))

(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM) PY (cadar PPM)
            X (fix (/ (+ (car PX) (car PY)) 2))
            Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
            P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (> (last (car PPM)) 0)
            (while (< (last (car PPM)) 1)
                   (setq PPM (cdr PPM)
                         LLL/LLL (cdr LLL/LLL)
                         LLL (car LLL/LLL)
                         PX (caar PPM) PY (cadar PPM)
                         X (fix (/ (+ (car PX) (car PY)) 2))
                         Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
                         N (ELEMENT **9**9**)
                         **V**V** (COMMUTA **V**V**)
                         SS (ssget "_C" PX PY))
                   (V+- "VAR-1")))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (setq GR N)
        (if (not LLL) (INI-LLL L1A9))
        (MASCARA (ELEMENT LLL))
        (setq N GR PPM (cons (list PX PY M) PPM)
              LLL (ACTUALITZA **V**V** N P LLL T) **V**V** (car LLL)
              LLL (cadr LLL) LLL/LLL (cons LLL LLL/LLL)))))
```

No allargarem això reproduint les funcions TECLAT, FES-SOLUCIO, CAN->VAR, 2-9V, SUD-MIN, FES-SUDOKU-2, FES-SUDOKU o C:SUDOKULUM, perquè l'afectació és mínima i ens limitarem a consignar-les: en TECLAT i 2-9V, MASC substitueix MASCARA tal i com detallàvem a la pàgina 469; FES-SOLUCIO perd C->F, que passa a variable local de MASCARA; a CAN->VAR, l'accés a F=3 incorpora T com a segon argument; de SUD-MIN s'han eliminat els arguments 9L (a l'unic accés que queda el seu valor és **V*V**) i PREVI (en aquest únic accés val nil), permetent la segona supressió substituir la condició composta (or (atom E) (and PREVI (equal E (list X Y)))) per (atom E); a FES-SUDOKU-2, l'accés a SUD-MIN perd els arguments; a l'inici de FES-SUDOKU fem POST = T, quan (= 1-2 "1"); a C:SUDOKULUM (en el programa, doncs), ara hi ha les variables LLISTES, 0*0, NIL*NIL i TT (cap novetat, perquè sortien a la versió de referència), i desapareix NOPLUS (tan obsoleta a FES-SOLUCIO com a FES-SUDOKU-1).

Tanmateix, i malgrat ser un intent fallit (s'han aconseguit avenços, tot i que no tan definitius com preteníem), volem presentar el penúltim intent d'eixamplar el coll d'ampolla del programa: la funció COMPAT-PERFILS (l'últim, el definitiu i més dràstic, s'aborda en el capítol *Consells practics per evitar una executio prècox*). Però, ¿per què justament ara ens ha donat per tractar d'escurçar de nou uns temps d'execució en què ja hi havíem esmerçat esforços considerables (capítols *Detectar solucions compatibles*, *Solucions compatibles ho són totes les que hi són...* i *... Però no hi eren totes les que ho són*), sense obtenir resultats massa brillants? Doncs perquè ara necessitàvem veure com responia FES-SUDOKU-2 en ser aplicat a casos potser conflictius (per la presència de configuracions AAA-AAA i/o AAA-BBB) com els considerats a la primera part del capítol *Etapa prèvia: crear una solució, IV (... o a Camprodon)*, per tal de decidir si la incorporació en ANORMALS d'algun dispositiu semblant a TERCETS + FILS/COLS era tan imperativa com en FES-SOLUCIÓ, i gairebé sempre ens hem trobat que, en activar POST la funció SUD-MIN, l'execució de COMPAT-PERFILS es feia més llarga que un dia sense pa, impedint-nos de fet la verificació. Prenent com a referència el sudoku "difícil" de LA VANGUARDIA del dia 25-02-08, ja utilitzat en el segon dels tres capítols esmentats i que emplenarem en el mateix ordre, ens permetrem recordar que havíem aconseguit abaixar els dos temps d'espera principals (determinació de la quantitat de solucions normalitzades compatibles amb la part pública del sudoku, en activar-se POST, i de la quantitat de solucions compatibles no normalitzades, en quedar només una de normalitzada), en acabar aquest capítol, a 8'50" + 11'40" = 20'30". Doncs bé: sense haver fet res aparentment, podem anunciar que amb les versions precedents i actual del programa la segona espera es redueix trenta-cinc vegades, passant a 8'50" + 0'20" = 9'10".

A què és degut el miracle? Sens dubte, a la millora de la funció ACTUALITZA, que forma part de l'expressió autorecursiva de RE+MEMBER < ANORMALS < FES-SUDOKU-2:

```
        ....................................................
        (if (or PRE (not NORM))
            (progn
              (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
              (foreach E L1A9
                (if (member E R-N)
                    (progn
                      (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                      (RE+MEMBER (car 2R) (cadr 2R))))))
            (if NORM (setq PRE T))))))
```

En concret, el fet que amb REAL = nil també funcioni ACTUALITZA-PLUS (possibilitat que d'entrada havíem exclòs, en creure erròniament que alentiria les exploracions a càrrec de RE-MEMBER i RE+MEMBER) i l'eficiència creixent d'aquesta funció en són els responsables. De tota manera, els gairebé nou minuts que triguen PERFIL-V*V i COMPAT-PERFILS en arribar a la conclusió que hi ha 1.117 solucions normalitzades compatibles amb la part pública de V. E. NORMALITZADA en activar-se POST, no dóna peu a gens d'autocomplaença, en tractar-se d'un exercici pensat acuradament perquè POST saltés quan teníem 18 caselles activades (sense comptar amb 2 forats tapats). És per això que l'autor, tot i no venir-li gens de gust desempolsar la feina feta dos anys enrera, s'hi va posar i, gratant, gratant, encara va poder reduir aquest temps a poc més de la tercera part (de 8'50" a 3'30", i després encara a 3'10"), però el guany de velocitat, per bé que pogués resultar prou satisfactori en termes relatius, seguia sent molt magre en termes absoluts. Abans de seguir, oferirem el desglossament dels temps (tot en negreta, però, per no confondre ningú que passés fulls ràpidament, sobre la condició d'annex d'aquests continguts), i no només per satisfer la curiositat d'eventuals lectors inassequibles al desànim, sinó perquè ens ajudarà força a identificar on progressem adequadament i en què cal millorar.

Versió heretada de PERFIL-V*V i COMPAT-PERFILS:

0:00 ***Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 196.***
.....
0:05 ***Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 96.***

0:55 ***Entre 196 línies compatibles amb la 1ª, 76 amb la 2ª i 324 amb la 3ª,***
hi ha 10232 tríades normalitzades compatibles amb la V. E. NORMALITZADA.
5:50 ***Entre 628 línies compatibles amb la 4ª, 340 amb la 5ª i 168 amb la 6ª,***
hi ha 1920 tríades normalitzades compatibles amb la V. E. NORMALITZADA.
5:55 ***Entre 92 línies compatibles amb la 7ª, 40 amb la 8ª i 96 amb la 9ª,***
hi ha 2496 tríades normalitzades compatibles amb la V. E. NORMALITZADA.

8:50 ***Entre les 10232 tríades 1-2-3, les 1920 tríades 4-5-6 i les 2496 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 1117.***

Versió millorada I de PERFIL-V*V i COMPAT-PERFILS:

0:00 ***Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 196.***
.....
0:05 ***Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 96.***

0:15 ***Entre 196 línies compatibles amb la 1ª, 76 amb la 2ª i 324 amb la 3ª,***
hi ha 10232 tríades normalitzades compatibles amb la V. E. NORMALITZADA.
0:35 ***Entre 628 línies compatibles amb la 4ª, 340 amb la 5ª i 168 amb la 6ª,***
hi ha 1920 tríades normalitzades compatibles amb la V. E. NORMALITZADA.
0:35 ***Entre 92 línies compatibles amb la 7ª, 40 amb la 8ª i 96 amb la 9ª,***
hi ha 2496 tríades normalitzades compatibles amb la V. E. NORMALITZADA.

3:30 ***Entre les 10232 tríades 1-2-3, les 1920 tríades 4-5-6 i les 2496 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 1117.***

Versió millorada II de PERFIL-V*V i COMPAT-PERFILS:

0:00 ***Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 196.***
.....
0:05 ***Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 96.***

0:15 ***Entre 196 línies compatibles amb la 1ª, 76 amb la 2ª i 324 amb la 3ª,***
hi ha 10232 tríades normalitzades compatibles amb la V. E. NORMALITZADA.
0:35 ***Entre 628 línies compatibles amb la 4ª, 340 amb la 5ª i 168 amb la 6ª,***
hi ha 1920 tríades normalitzades compatibles amb la V. E. NORMALITZADA.
0:35 ***Entre 92 línies compatibles amb la 7ª, 40 amb la 8ª i 96 amb la 9ª,***
hi ha 2496 tríades normalitzades compatibles amb la V. E. NORMALITZADA.

3:10 ***Entre les 10232 tríades 1-2-3, les 1920 tríades 4-5-6 i les 2496 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 1117.***

(Si a algú se li ha ocorregut comparar aquests desglossaments amb els del capítol precedent, *... Però no hi eren totes les que ho són*, de seguida se n'haurà adonat que els temps els expressàvem en HORES:MINUTS, perquè en l'etapa de compatibilitat entre els elements dels tres conjunts de tríades de línies sovint es trigava força hores -això si l'ordinador no es penjava per ofegament de memòria-, mentre que ara ho hem fet en MINUTS:SEGONS.)

Més que el detall les xifres, és interessant de conèixer com hem aconseguit passar de la versió heretada a la versió millorada I; però el pas d'aquesta a la versió millorada II és força més feixuc (10 temptatives més no van reixir a rebaixar-ne els temps), de forma que només donarem el codi complet de les versions esmentades, limitant-nos a una visió panoràmica de la resta, il·lustrada puntualment amb codi.

La innovació transcendental ha estat complementar LINIA-V*V (text compost per 9 caràcters numèrics) amb KLINIA-V*V (llista formada per 3 subllistes de 3 caràcters numèrics cadascuna), i LINIES-V*V (llista de 9 subllistes formades per LINIA-V*Vs, corresponents a les files 1ª, 2ª... 9ª) amb KLINIES-V*V (llista de 9 subllistes formades per KLINIA-V*Vs, corresponents a les files 1ª, 2ª... 9ª), i substituir les funcions COMPAT-2L i COMPAT-3L (que treballaven amb LINIA-V*Vs) per KOMPAT-2L i KOMPAT-3L (que treballen amb KLINIA-V*Vs). D'aquesta forma, la determinació dels requadres 9×3 amb les tres files (files compatibles amb les homòlogues de la V. E. NORMALITZADA) compatibles entre si i, en conseqüència, amb la V. E. NORMALITZADA, s'alleugerirà força, perquè no és el mateix avançar caràcter a caràcter per dos textos de nou fins a detectar una eventual coincidència (fent ús de les funcions substr i wcmatch) que fer-ho comparant llistes de caràcters (mitjançant member). Tanmateix, una vegada obtingut un guany de 5'50" - 0'30" = 5'20" en aquesta etapa (la segona de les tres enunciades en el capítol *Detectar solucions compatibles*, just després d'inventariar en PERFIL-V*V les files primeres, segones... novenes compatibles amb les caselles activades), veureu que prescindim de KLINIES-V*V i seguim treballant amb la mateixa estructura de dades que en la versió precedent. Però abans de seguir amb la discussió de les possibilitats alternatives per a la tercera etapa (inventariar aquelles combinacions de tríades compatibles que també siguin compatibles entre si), aquí teniu les modificacions introduïdes en el codi:

```
(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V KLINIA-V*V KKE KE)
  (if (not L1) (C:CARREGA-123456789))
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
  (repeat 3
    (setq J (- J 9) K (+ K 3))
    (repeat 3
      (setq J (+ J 3) K (- K 3) W ())
      (repeat 3
        (setq J (- J 3) K (1+ K))
        (repeat 3
          (setq J (1+ J) V (nth J (nth K **V*V**)))
          (if (atom V) (setq W (cons V W)))))
      (cond ((= J 2) (setq A W))
            ((= J 5) (setq B W))
            (T       (setq C W))))
    (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
  (setq L1C (reverse L1C) J -1)
  (repeat 9
    (setq J (1+ J) K -1 L ())
    (repeat 9 (setq K (1+ K) C (nth J (nth K **V*V**))
                    L (if (atom C) (cons C L) L)))
    (setq L2C (cons L L2C)))
  (setq L2C (reverse L2C) LINIES-V*V () KLINIES-V*V () K -1)
  (terpri)
  (repeat 9
    (setq LINIA-V*V () KLINIA-V*V () L "" K (1+ K))
    (mapcar '(lambda (E F G)
               (setq L (strcat L (if (or F G)
                                     (if (atom E)
                                         (itoa E)
                                         (progn
                                           (setq C "]")
                                           (foreach H (append F G)
                                             (setq C (strcat (itoa H) C)))
                                           (strcat "[~" C)))
                                     "#"))))
            (nth K **V*V**) (nth K L1C) L2C)
    (cond ((= K 0) (setq A "123456789" B "198765432"))
          ((= K 1) (setq A "213456789" B "897654321"))
          ((= K 2) (setq A "312456789" B "987654321"))
          ((= K 3) (setq A "213456789" B "498765321"))
          ((= K 4) (setq A "312456789" B "897654321"))
          ((= K 5) (setq A "412356789" B "987654321"))
          ((= K 6) (setq A "312456789" B "798654321"))
          ((= K 7) (setq A "412356789" B "897654321"))
          (T       (setq A "512346789" B "987654321")))
```

```
    (foreach E (reverse (member B (reverse (member A L1))))
      (if (wcmatch E L)
          (progn
            (setq LINIA-V*V (cons E LINIA-V*V) KKE () *K* 10)
            (repeat 3
              (setq KE ())
              (repeat 3 (setq *K* (1- *K*) KE (cons (substr E *K* 1) KE)))
              (setq KKE (cons KE KKE)))
            (setq KLINIA-V*V (cons KKE KLINIA-V*V)))))
    (setq LINIES-V*V (cons (reverse LINIA-V*V) LINIES-V*V)
          KLINIES-V*V (cons (reverse KLINIA-V*V) KLINIES-V*V)
          *K* (itoa (length LINIA-V*V)) *KK* (cons *K* *KK*))
    (terpri)
    (prompt (strcat "\nEntre 362880 línies, de compatibles amb la " (itoa (1+ K))
                    "ª de la V. E. NORM. n'hi ha " *K* "."))
  (setq LINIES-V*V (reverse LINIES-V*V) KLINIES-V*V (reverse KLINIES-V*V)
        L1 () *KK* (reverse *KK*))
  (terpri))

(defun KOMPAT-2L ()
  (setq OK T)
  (mapcar '(lambda (L0 L1)
             (foreach E L1 (if (and OK (member E L0)) (setq OK ()))))
          KT0 KT1)
  (if OK (setq KT01 (list (append (cadr KT0) (cadr KT1))
                          (append (last KT0) (last KT1))))))

(defun KOMPAT-3L ()
  (setq OK T)
  (mapcar '(lambda (L01 L2)
             (foreach E L2 (if (and OK (member E L01)) (setq OK ()))))
          KT01 (cdr KT2))
  OK)

(defun COMPAT-PERFILS (/ L1 LL0 KLL0 LL1 KLL1 L2 LL2 KLL2 KT01 3L M N TERCETS-V*V
                        CC0 CC1 CC2 CCC1 CCC2 TT0 TT1 TT2 LOCT1 LOCT2 TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  (repeat 9 (setq K (1+ K) TOT (cons (strcat "*" (itoa K) "*") TOT)))
  (repeat 3
    (setq LL0 (car LINIES-V*V) LL1 (cadr LINIES-V*V) LL2 (caddr LINIES-V*V)
          LINIES-V*V (cdddr LINIES-V*V) TERCETS-V*V ()
          KLL0 (car KLINIES-V*V) KLL1 (cadr KLINIES-V*V) KLL2 (caddr KLINIES-V*V)
          KLINIES-V*V (cdddr KLINIES-V*V))
    (mapcar '(lambda (T0 KT0)
               (mapcar '(lambda (T1 KT1)
                          (if (and (< T0 T1) (KOMPAT-2L))
                              (mapcar '(lambda (T2 KT2)
                                         (if (and (< T1 T2) (KOMPAT-3L))
                                             (setq TERCETS-V*V
                                                   (cons (list T0 T1 T2)
                                                         TERCETS-V*V))))
                                      LL2 KLL2)))
                        LL1 KLL1))
            LL0 KLL0)
    (if TERCETS-V*V (setq *T* (itoa (length TERCETS-V*V))
                          *TT* (append *TT* (list *T*))
                          SUDOKUS-V*V (append SUDOKUS-V*V
                                              (list (reverse TERCETS-V*V)))))
    (prompt (strcat "\nEntre " (nth *K* *KK*) " línies compatibles amb la "
                    (itoa (setq *K* (1+ *K*))) "ª, " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª i " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª,\nhi ha " *T* " tríades "
                    "normalitzades compatibles amb la V. E. NORMALITZADA.\n")))
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq KLINIES-V*V () TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
```

```
        (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
        (foreach T0 TT0
          (if (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0)
                    L1 (SUPR (substr (cadr T0) 1 1) TOT)
                    L1 (SUPR (substr (last T0) 1 1) L1)
                    PRIM1 (cdr (assoc (car L1) LOCT1)))
              (mapcar '(lambda (T1 CC1)
                         (setq L2 (cdr L1) K -1 OK T)
                         (while (and OK (< (setq K (1+ K)) 9))
                                (if (wcmatch (nth K CC1) (nth K CC0))
                                    (setq OK ())))
                         (if (and OK (setq CC1 (PERF-COLS CC1)
                                           L2 (SUPR (substr (cadr T1) 1 1) L2)
                                           L2 (SUPR (substr (last T1) 1 1) L2)
                                           PRIM2 (cdr (assoc (car L2) LOCT2))))
                             (mapcar '(lambda (T2 CC2)
                                        (setq K -1 OK T)
                                        (while (and OK (< (setq K (1+ K)) 9))
                                               (if (or (wcmatch
                                                         (setq J (nth K CC2))
                                                         (nth K CC0))
                                                       (wcmatch J (nth K CC1)))
                                                   (setq OK ())))
                                        (if OK (setq SUDOKUS-V*V
                                                     (cons (SUD-NUM) SUDOKUS-V*V)
                                                     S-V*V (cons T S-V*V))))
                                     (member (nth PRIM2 TT2) TT2)
                                     (MEMBRE PRIM2 CCC2))))
                       (member (nth PRIM1 TT1) TT1)
                       (MEMBRE PRIM1 CCC1))))
        (setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
              KK (length S-V*V) S-V*V (list (cons P S-V*V)))
        (prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                                       (nth 1 *TT*) " tríades 4-5-6 i les "
                                       (nth 2 *TT*) " tríades 7-8-9,"
                        "\nde solucions normalitzades compatibles amb la V. E. "
                        "NORMALITZADA n'hi ha " (itoa KK) "."))
        (textscr)
        (SEGUIR ())
        (graphscr))))
```

Mostrat el codi del que 3 pàgines enrera anomenàvem versió millorada I (pàg. 474), cosa que convenia fer per no dispersar-nos, recuperem el fil de la discussió amb una pregunta que gairebé és inevitable després de l'última afirmació: ¿per què no s'intentava abordar la tercera part del procés amb una estructura de dades hereva de KLINIES-V*V, en lloc de ser-ho de LINIES-V*V? Doncs sí que es va intentar, però els resultats no superaven els d'aquesta solució i, sense retrocedir als temps de l'anterior, s'hi acostaven força. Les 2 variants (4 de fet, com veureu de seguida) es basaven en el mateix: modificació de COLS-3L, que en comptes de muntar llistes de 9 elements, cada un dels quals era un text format per 3 caràcters d'un requadre 9×3 alineats verticalment, les munta de manera que els elements són subllistes de 3 caràcters numèrics o dels valors enters corresponents; supressió de PERF-COLS, de manera que CC0 es quedava amb el valor (COLS-3L T0) al llarg de tot el procés, CC1 amb el valor (COLS-3L T1) i CC2 amb el valor (COLS-3L T2), i substitució de les comparacions

```
  (while (and OK (< (setq K (1+ K)) 9))
         (if (wcmatch (nth K CC1) (nth K CC0))
             (setq OK ())))
```

i

```
  (while (and OK (< (setq K (1+ K)) 9))
                 (if (or (wcmatch (setq J (nth K CC2)) (nth K CC0))
                         (wcmatch J (nth K CC1)))
                     (setq OK ())))
```

per les dues fórmules alternatives que detallarem. Comencem per les dues versions de la funció COLS-3L, refoses en una de sola (en realitat, l'únic que cal excloure per obtenir llistes de caràcters numèrics i no de valors enters és el subratllat):

```
(defun COLS-3L (I / C CC)
  (setq K 10)
  (repeat 9
    (setq K (1- K) C ())
    (foreach J (reverse I) (setq C (cons (atoi (substr J K 1)) C)))
  (setq CC (cons C CC))))
```

Pel que fa a les dues comparacions amb wcmatch, de primer vam substituir-les per

```
  (while (and OK (< (setq K (1+ K)) 9))
         (setq C0 (nth K CC0))
         (foreach E (nth K CC1)
           (if (and OK (member E C0))
               (setq OK ()))))
```

i

```
  (while (and OK (< (setq K (1+ K)) 9))
         (setq C0 (nth K CC0) C1 (nth K CC1))
         (foreach E (nth K CC2)
           (if (and OK (or (member E C0) (member E C1)))
               (setq OK ()))))
```

amb el resultat decebedor que hem esmentat.

Alternativament, per si aconseguíem algun guany significatiu de temps refonent CC0 i CC1 en una sola llista CC01, fent

```
  (setq CC01 (mapcar '(lambda (C0 C1) (append C0 C1)) CC0 CC1)
```

per processar-les mitjançant una única funció member, vam transformar la segona expressió d'aquesta manera

```
  (while (and OK (< (setq K (1+ K)) 9))
         (setq C01 (nth K CC01))
         (foreach E (nth K CC2)
           (if (and OK (member E C01))
               (setq OK ()))))
```

Guanyàvem uns segons respecte la precedent però, tot plegat, misèria i companyia. (Reteniu tanmateix aquesta tàctica de refondre CC0 i CC1, perquè l'única variant amb què aconseguirem millorar el temps 3'10", que presentarem un cop finalitzada aquesta exposició de fracassos), recull la idea però mantenint l'eina wcmatch.)

En vista d'això, vam provar en una altra direcció: evitar que les dades de què cal disposar reiteradament s'hagin de recalcular cada vegada, obtenint-les i guardant-les d'una vegada per totes. En aquesta línia, vam provar 10 variants, amb fortuna desigual però sempre amb resultats situats per sota dels de "Versió millorada I".

Un intent va consistir a evitar que, amb cada nou requadre 9×3 T1 de TT1 però amb uns mateixos requadres T0 de TT0 i T2 de TT2, s'hagués de repetir la comprovació (wcmatch (nth K CC2) (nth K CC0)), estesa a 9 iteracions, com mostra el subratllat

```
  (setq K -1 OK T)
  (while (and OK (< (setq K (1+ K)) 9))
         (if (or (wcmatch (setq J (nth K CC2)) (nth K CC0))
                 (wcmatch J (nth K CC1)))
             (setq OK ())))
```

Amb aquest propòsit vam dotar a cada T0 d'una llista d'associacions T-02, homòloga de (member (nth PRIM1 TT1) TT1) i de (MEMBRE PRIM1 CCC1), en què l'índex de cada element era l'ordinal N de la llista i la part referenciada era T o nil, segons el valor de l'esmentada expressió (wcmatch (nth K CC0) (nth K CC1)). Un inconvenient era que, en fer-se sobre la marxa, abans d'usar la part referenciada calia veure si ja existia l'element N-èssim (l'índex N l'inicialitzàvem a (1- PRIM2)), fent (if (setq ... WC (assoc N T-02)) ...). Reproduint-ne únicament el rovell de l'ou,

```
  ....................................................................
  (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
  (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
  (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
  (foreach T0 TT0
    (if (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0)
              L1 (SUPR (substr (cadr T0) 1 1) TOT)
              L1 (SUPR (substr (last T0) 1 1) L1) T-02 ()
              PRIM1 (cdr (assoc (car L1) LOCT1)))
        (mapcar '(lambda (T1 CC1)
                   (setq L2 (cdr L1) K -1 OK T)
```

```
                    (while (and OK (< (setq K (1+ K)) 9))
                           (if (wcmatch (nth K CC1) (nth K CC0))
                               (setq OK ())))
                    (if (and OK (setq CC1 (PERF-COLS CC1)
                                      L2 (SUPR (substr (cadr T1) 1 1) L2)
                                      L2 (SUPR (substr (last T1) 1 1) L2)
                                      PRIM2 (cdr (assoc (car L2) LOCT2))
                                      N (if PRIM2 (1- PRIM2))))
                        (mapcar '(lambda (T2 CC2)
                                   (if (setq OK T K -1 N (1+ N) WC (assoc N T-02))
                                       (if (cdr WC)
                                           (setq OK ())
                                           (while (and OK (< (setq K (1+ K)) 9))
                                                  (if (wcmatch (nth K CC2)
                                                               (nth K CC1))
                                                      (setq OK ()))))
                                       (progn
                                         (setq WC ())
                                         (while (and OK (< (setq K (1+ K)) 9))
                                                (if (or (if (wcmatch
                                                              (setq J (nth K CC2))
                                                              (nth K CC0))
                                                            (setq WC T))
                                                        (wcmatch J (nth K CC1)))
                                                    (setq OK ())))
                                         (if (or WC (and (not WC) (> K 7)))
                                             (setq T-02 (cons (cons N WC) T-02))))))
                                 (if OK (setq SUDOKUS-V*V
                                              (cons (SUD-NUM) SUDOKUS-V*V)
                                              S-V*V (cons T S-V*V))))
                                 (member (nth PRIM2 TT2) TT2)
                                 (MEMBRE PRIM2 CCC2))))
                (member (nth PRIM1 TT1) TT1)
                (MEMBRE PRIM1 CCC1))))
                ..................................................................
```

Una limitació era que els elements de T-02 no sempre quedaven definits la primera vegada que s'hi accedia, depenent de l'element de comparació CC1: quan posàvem T, de segur que hi havia algun valor K per al qual (wcmatch (nth K CC2) (nth K CC0)) era T; però si OK s'activava perquè (wcmatch (nth K CC2) (nth K CC1)) era T, tant podia ser que (wcmatch (nth K CC2) (nth K CC0)) hagués estat T per a algun valor K posterior (si haguéssim seguit comparant) o que fos nil per a tots els K, raó per la qual deixàvem T-02 igual i repetíem la comprovació amb tots els nous requadres 9×3 T1 que s'anessin presentant. En definitiva, només hi posàvem nil si hi havia hagut ocasió d'arribar a K = 8, amb (wcmatch (nth K CC2) (nth K CC0)) valent nil.

Una manera senzilla d'evitar això va ser forçar que en la primera comparació amb un requadre T1 s'arribés a K = 8, tret que abans (wcmatch (nth K CC2) (nth K CC0)) fos T, correcció amb què vam augmentar una mica l'eficiència. Com que la correcció es concentra en l'expressió (if (setq OK T K -1 N (1+ N) WC (assoc N T-02)) ...), ens limitarem a reproduir el codi d'aquesta part, precedit de la repetició del de la versió precedent (utilitzant ara l'amplada disponible, per no fragmentar-les):

```
  ..............................................
  (if (setq OK T K -1 N (1+ N) WC (assoc N T-02))
      (if (cdr WC)
          (setq OK ())
          (while (and OK (< (setq K (1+ K)) 9))
                 (if (wcmatch (nth K CC2) (nth K CC1))
                     (setq OK ()))))
      (progn
        (setq WC ())
        (while (and OK (< (setq K (1+ K)) 9))
               (if (or (if (wcmatch (setq J (nth K CC2)) (nth K CC0))
                           (setq WC T))
                       (wcmatch J (nth K CC1)))
                   (setq OK ())))
        (if (or WC (and (not WC) (> K 7)))
            (setq T-02 (cons (cons N WC) T-02)))))
            ....................................
```

```
  ..........................................
  (if (setq OK T K -1 N (1+ N) WC (assoc N T-02))
      (if (cdr WC)
          (setq OK ())
          (while (and OK (< (setq K (1+ K)) 9))
                      (if (wcmatch (nth K CC2) (nth K CC1))
                          (setq OK ()))))
      (progn
        (setq WC ())
        (while (and (not WC) (< (setq K (1+ K)) 9))
               (if (or (if (wcmatch (setq J (nth K CC2)) (nth K CC0))
                           (setq WC T))
                       (wcmatch J (nth K CC1)))
                   (setq OK ())))
        (setq T-02 (cons (cons N WC) T-02))))
        ..................................
```

Però l'eficàcia de la llista T-02 va augmentar força més formant-la prèviament:

```
  ....................................................................
  (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
  (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
  (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
  (foreach T0 TT0
    (setq CC0 (COLS-3L T0) CC0 (PERF-COLS CC0) T-02 ())
    (mapcar '(lambda (T2 CC2)
               (setq K -1 OK T)
               (while (and OK (< (setq K (1+ K)) 9))
                      (if (wcmatch (nth K CC2) (nth K CC0)) (setq OK ())))
               (setq T-02 (cons (not OK) T-02)))
            TT2 CCC2)
    (if (setq T-02 (reverse T-02)
              L1 (SUPR (substr (cadr T0) 1 1) TOT)
              L1 (SUPR (substr (last T0) 1 1) L1)
              PRIM1 (cdr (assoc (car L1) LOCT1)))
        (mapcar '(lambda (T1 CC1)
                   (setq L2 (cdr L1) K -1 OK T)
                   (while (and OK (< (setq K (1+ K)) 9))
                          (if (wcmatch (nth K CC1) (nth K CC0))
                              (setq OK ())))
                   (if (and OK (setq CC1 (PERF-COLS CC1)
                                     L2 (SUPR (substr (cadr T1) 1 1) L2)
                                     L2 (SUPR (substr (last T1) 1 1) L2)
                                     PRIM2 (cdr (assoc (car L2) LOCT2))))
                       (mapcar '(lambda (T2 CC2 WC)
                                  (if WC
                                     (setq OK ())
                                     (progn
                                       (setq K -1 OK T)
                                       (while (and OK (< (setq K (1+ K)) 9))
                                              (if (wcmatch (nth K CC2)
                                                           (nth K CC1))
                                                  (setq OK ()))))))
                                  (if OK (setq SUDOKUS-V*V
                                               (cons (SUD-NUM) SUDOKUS-V*V)
                                               S-V*V (cons T S-V*V))))
                               (member (nth PRIM2 TT2) TT2)
                               (MEMBRE PRIM2 CCC2)
                               (MEMBRE PRIM2 T-02))))
                (member (nth PRIM1 TT1) TT1)
                (MEMBRE PRIM1 CCC1))))
                ..................................
```

I encara més, creant, a més de la llista T-02, la llista TT-12 de subllistes T-12:

```
  ....................................................................
  (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
  (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
  (foreach T1 (reverse TT1)
    (setq CC1 (COLS-3L T1) PCC1 (PERF-COLS CC1) T-12 ())
```

```
    (mapcar '(lambda (T2 CC2)
               (setq K -1 OK T)
               (while (and OK (< (setq K (1+ K)) 9))
                      (if (wcmatch (nth K CC2) (nth K PCC1)) (setq OK ())))
               (setq T-12 (cons (not OK) T-12)))
            TT2 CCC2)
    (setq TT-12 (cons (reverse T-12) TT-12) CCC1 (cons CC1 CCC1)))
  (foreach T0 TT0
    (if (setq CC0 (COLS-3L T0) PCC0 (PERF-COLS CC0)
              L1 (SUPR (substr (cadr T0) 1 1) TOT)
              L1 (SUPR (substr (last T0) 1 1) L1) T-02 ()
              PRIM1 (cdr (assoc (car L1) LOCT1)))
        (progn
          (mapcar '(lambda (T2 CC2)
                     (setq K -1 OK T)
                     (while (and OK (< (setq K (1+ K)) 9))
                            (if (wcmatch (nth K CC2) (nth K PCC0)) (setq OK ())))
                     (setq T-02 (cons (not OK) T-02)))
                  TT2 CCC2)
          (setq T-02 (reverse T-02))
          (mapcar '(lambda (T1 CC1 T-12)
                     (setq L2 (cdr L1) K -1 OK T)
                     (while (and OK (< (setq K (1+ K)) 9))
                            (if (wcmatch (nth K CC1) (nth K PCC0)) (setq OK ())))
                     (if (and OK (setq L2 (SUPR (substr (cadr T1) 1 1) L2)
                                       L2 (SUPR (substr (last T1) 1 1) L2)
                                       PRIM2 (cdr (assoc (car L2) LOCT2))))
                         (mapcar '(lambda (T02 T12)
                                    (if (not (or T02 T12))
                                        (setq SUDOKUS-V*V
                                              (cons (SUD-NUM) SUDOKUS-V*V)
                                              S-V*V (cons T S-V*V))))
                                 (MEMBRE PRIM2 T-02)
                                 (MEMBRE PRIM2 T-12))))
                  (member (nth PRIM1 TT1) TT1)
                  (MEMBRE PRIM1 CCC1)
                  (MEMBRE PRIM1 TT-12)))))
                  ....................................
```

Amb la discussió de 7 variants creiem que hem donat prou pistes al lector perquè, si s'anima a assajar altres camins, no perdi més temps amb aquests. De les 3 que queden ens limitarem a consignar que dues s'incrivien en l'òrbita de les quatre darreres i eren intents de simplificar-les: una pretenia alleujar la primera del grup fent que T-02 només enregistrés valors T, cosa que permetia prescindir de la llista d'associacions (T-02 quedava reduïda a una llista amb els ordinals dels requadres 9×3 en què (wcmatch (nth K CC2) (nth K CC0)) fos T) però, com hauríem pogut preveure, la durada es va multiplicar per 3; l'altra pretenia millorar la segona, en el sentit d'evitar que s'anés repetint el recompte (setq N (1+ N)) per poder conèixer l'ordinal de T2 i CC2 (i amb aquest propòsit cada requadre T2 de la llista TT2 anava precedit d'un enter indicant la seva posició a la llista), però els temps no van experimentar cap variació. L'última només va aconseguir igualar el temps de la versió millorada I, però a costa de complicar-la considerablement: es pretenia afinar les posicions LOCT1 i LOCT2 eliminant les adreces corresponents als caràcters inicials de les primeres i segones tríades en què totes les segones o terceres files comencessin amb el mateix caràcter, però es va veure que aquest esforç era estèril perquè almenys en l'exemple de referència això ja era tingut en compte en seleccionar les línies compatibles; s'hagués hagut d'aprofundir, per saber del cert si això passava sempre (dit en altres paraules, si l'absència d'un determinat caràcter a l'inici de la segona o tercera fila d'un tercet només podia esdevenir-se si apareixia a l'inici d'alguna altra fila o entre els tres primers caràcters de les altres dues files de la tríada, circumstàncies ben previstes) o trobàvem algun contraexemple que demostrés que una cosa no pressuposava l'altra, però aquesta línia va quedar en via morta en veure que tot plegat era més senzill.

Després d'aquestes cabòries, veiem per on va sortir la versió millorada II, única amb què hem aconseguit esgarrapar els temps de la versió millorada I. Partint de la tercera variant comentada, on refoníem les subllistes C0 i C1 de 3 caràcters numèrics (o valors enters) en subllistes C01 de 6, cada 9 de les quals formava una

llista CC01 amb el contingut de les 9 columnes dels dos primers requadres 9×3, vam decidir deixar aquestes llistes i la funció member, i tornar a jugar amb llistes de textos (cadenes de caràcters) i amb la funció wcmatch. De fet, no hauria calgut el circumloqui de les tres primeres variants (substituir els 9 textos de CC0, CC1 i CC2 per subllistes formades per 3 caràcters o enters) per adonar-nos (ja en la mateixa versió millorada I) que, si en la segona comparació entre aquests textos els primers arguments de les wcmatch complementàries (els textos a comparar) eren iguals, aquestes dues funcions es podien refondre en una que tingués com a segon argument (perfil o patró de comparació) l'unió dels primitius segons arguments, és a dir, que l'expressió

```
  (while (and OK (< (setq K (1+ K)) 9))
         (if (or (wcmatch (setq J (nth K CC2)) (nth K CC0))
                 (wcmatch J (nth K CC1)))
             (setq OK ())))
```

en què

```
  (setq CC0 (COLS-3L T0) CC0 (mapcar '(lambda (C0) (strcat "*[" C0 "]*")) CC0))
```

i

```
  (setq CC1 (COLS-3L T1) CC1 (mapcar '(lambda (C1) (strcat "*[" C1 "]*")) CC1))
```

era equivalent a

```
  (while (and OK (< (setq K (1+ K)) 9))
         (if (wcmatch (nth K CC2) (nth K CC01))
             (setq OK ())))
```

en què

```
  (setq CC0 (COLS-3L T0) CC1 (COLS-3L T1)
        CC01 (mapcar '(lambda (C0 C1) (strcat "*[" C0 C1 "]*")) CC0 CC1)
```

A la versió millorada II de COMPAT-PERFILS, que reproduïm íntegrament, distingirem entre les variables utilitzades per representar textos (CC0, CC1 i CC2) i les que representen perfils (PCC0 i PCC01), per evitar interferències (com ja fèiem en la 7ª variant comentada), i ometrem la funció PERF-COLS perquè aquí la seva actuació no és idèntica per muntar PCC0 o PCC01, i preferim explicitar-la en ambdós casos:

```
(defun COMPAT-PERFILS (/ L1 LL0 KLL0 LL1 KLL1 L2 LL2 KLL2 KT01 3L M N TERCETS-V*V
                         CC0 PCC0 CC1 PCC01 CC2 CCC1 CCC2 TT0 TT1 TT2 LOCT1 LOCT2
                         TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  (repeat 9 (setq K (1+ K) TOT (cons (strcat "*" (itoa K) "*") TOT)))
  (repeat 3
    (setq LL0 (car LINIES-V*V) LL1 (cadr LINIES-V*V) LL2 (caddr LINIES-V*V)
          LINIES-V*V (cdddr LINIES-V*V) TERCETS-V*V ()
          KLL0 (car KLINIES-V*V) KLL1 (cadr KLINIES-V*V) KLL2 (caddr KLINIES-V*V)
          KLINIES-V*V (cdddr KLINIES-V*V))
    (mapcar '(lambda (T0 KT0)
               (mapcar '(lambda (T1 KT1)
                          (if (and (< T0 T1) (KOMPAT-2L))
                              (mapcar '(lambda (T2 KT2)
                                         (if (and (< T1 T2) (KOMPAT-3L))
                                             (setq TERCETS-V*V
                                                   (cons (list T0 T1 T2)
                                                         TERCETS-V*V))))
                                      LL2 KLL2)))
                        LL1 KLL1))
            LL0 KLL0)
    (if TERCETS-V*V (setq *T* (itoa (length TERCETS-V*V))
                          *TT* (append *TT* (list *T*))
                          SUDOKUS-V*V (append SUDOKUS-V*V
                                              (list (reverse TERCETS-V*V)))))
    (prompt (strcat "\nEntre " (nth *K* *KK*) " línies compatibles amb la "
                    (itoa (setq *K* (1+ *K*))) "ª, " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª i " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª,\nhi ha " *T* " tríades "
                    "normalitzades compatibles amb la V. E. NORMALITZADA.\n")))
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq KLINIES-V*V () TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
```

```
(foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
(foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
(foreach T0 TT0
  (if (setq CC0 (COLS-3L T0)
            PCC0 (mapcar '(lambda (C0) (strcat "*[" C0 "]*")) CC0)
            L1 (SUPR (substr (cadr T0) 1 1) TOT)
            L1 (SUPR (substr (last T0) 1 1) L1)
            PRIM1 (cdr (assoc (car L1) LOCT1)))
      (mapcar '(lambda (T1 CC1)
                 (setq L2 (cdr L1) K -1 OK T)
                 (while (and OK (< (setq K (1+ K)) 9))
                        (if (wcmatch (nth K CC1) (nth K PCC0))
                            (setq OK ())))
                 (if (and OK (setq PCC01
                                   (mapcar '(lambda (C0 C1)
                                              (strcat "*[" C0 C1 "]*"))
                                           CC0 CC1)
                                   L2 (SUPR (substr (cadr T1) 1 1) L2)
                                   L2 (SUPR (substr (last T1) 1 1) L2)
                                   PRIM2 (cdr (assoc (car L2) LOCT2))))
                     (mapcar '(lambda (T2 CC2)
                                (setq K -1 OK T)
                                (while (and OK (< (setq K (1+ K)) 9))
                                       (if (wcmatch (nth K CC2)
                                                    (nth K PCC01))
                                           (setq OK ())))
                                (if OK (setq SUDOKUS-V*V
                                             (cons (SUD-NUM) SUDOKUS-V*V)
                                             S-V*V (cons T S-V*V))))
                             (member (nth PRIM2 TT2) TT2)
                             (MEMBRE PRIM2 CCC2))))
             (member (nth PRIM1 TT1) TT1)
             (MEMBRE PRIM1 CCC1))))
(setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
      KK (length S-V*V) S-V*V (list (cons P S-V*V)))
(prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                              (nth 1 *TT*) " tríades 4-5-6 i les "
                              (nth 2 *TT*) " tríades 7-8-9,"
                "\nde solucions normalitzades compatibles amb la V. E. "
                "NORMALITZADA n'hi ha " (itoa KK) "."))
(textscr)
(SEGUIR ())
(graphscr))))
```

Com que ara la comparació de cada text CC2 amb els perfils PCC0 i PCC1 no es fa per separat sinó que aquests dos últims estan refosos en PCC01, que únicament es construeix una vegada, no té sentit plantejar-se muntar les llistes d'associacions que (sobre la marxa o en una operació prèvia) caracteritzava les variants 4ª a 7ª. I, per acabar, recordem que cal declarar la variable KLINIES-V*V en FES-SUDOKU-2.

Després d'haver dedicat deu pàgines a un ronyós estalvi de 20" en el mètode 2, no podríem tancar aquest apèndix sense deixar constància que, si en començar aquest capítol aventuràvem tímidament que el mètode 1 (el nou camí del títol) semblava més àgil que el 2, a hores d'ara la superioritat del primer sobre el segon ja és indiscutible (i ens estarem de balls de xifres, que prou marejat estarà el lector amb les que s'han prodigat fins aquí). No podia ser d'altra manera, perquè ja hem comentat que en l'exemple de referència no era només COMPAT-PERFILS que llastava el mètode 2 (on havíem aconseguit esgarrapar els ridículs 20") sinó RE+MEMBER < < ANORMALS, i que si havia passat de 11'40" a 0'20" era per l'eficiència creixent d'ACTUALITZA-PLUS < ACTUALITZA, funció aquesta que també beneficiava RE-MEMBER < < MASCA < MASCARA. Però, com que abans d'aquest comentari n'havíem fet un altre a propòsit de FES-SUDOKU-1 (final pàg. 469), on s'afirmava que ja no existia el punt de ruptura que marcava SUD-MIN i que els temps d'espera (menors que a la versió anterior) hi eren des del principi, potser era necessari subratllar que el balanç de conjunt era netament positiu i que de resultes d'això el mètode 1 es desmarcava del 2, quant a rendiment: els clics arrenquen amb una cadència de dos segons que es redueix paulatinament, però no es produeixen les llargues esperes del mètode 2.

****
****
****

Actualització a l'última versió de FES-SOLUCIO del capítol 7- *Etapa prèvia: crear una solució, V (la Seca, la Meca i la vall d'Andorra)*.

L'adaptació a la versió esmentada es limita a qüestions purament tècniques, perquè no introduïm cap canvi d'estratègia en relació a allò que hem establert en l'annex precedent: tant a FES-SOLUCIO com a FES-SUDOKU-1 la maquinària pesant RE-MEMBER es posarà en marxa des del primer moment, mentre que a FES-SUDOKU-2 el tàndem compost per PERFIL-V*V i COMPAT-PERFILS només intervindrà quan SUD-MIN ho hagi autoritzat.
Quant al nivell d'actuació d'ACTUALITZA-PLUS < ACTUALITZA, si el qualifiquem així,
1) detecció i emplenament automàtic de caselles de valor prefixat (forats),
2) detecció d'infraccions a la norma 1, avortant l'exploració,
3) detecció d'infraccions a la norma 2/3, avortant l'exploració,
4) detecció de configuracions AAA-AAA i AAA-BBB, explicitant tercets implícits, i
5) detecció de conjunts tancats de valors admissibles amb la conseqüent depuració,
(podem considerar que en el nivell 4 actuem específicament sobre un cas particular de la problemàtica genèricament abordada en el 5, que és el nou nivell introduït a la versió de referència, i que surt a compte fer-ho abans i amb un mètode *ad hoc*) resultarà menys farragós enumerar el grau d'aprofundiment practicat a cadascuna de les situacions següents, totes relatives a l'assignació d'un valor a una casella:
- A FES-SOLUCIO:
** - Exploració de tots els candidats (fins a trobar alguna solució compatible amb cadascun d'ells):**
**   - ACTUALITZA < RE-MEMBER < MASCARA < FES-SOLUCIO (REAL = nil):**
**   - INI = T: 1, 2, 3, 4 i 5.**
**   - INI = nil: 1 i 2.**
** - Elecció d'un candidat vàlid A-N i actualització:**
**   - ACTUALITZA < FES-SOLUCIO (REAL = T i INI = nil):**
**   - Si (assoc A-N TN/R-L) = T, no es repeteix l'execució d'ACTUALITZA-PLUS.**
**   - Si (assoc A-N TN/R-L) = nil: 1 i 2 (la inspecció 2 haurà estat supèrflua).**
- A FES-SUDOKU-1, quan s'activa una casella:
** - Exploració del candidat preestablert i de la resta de candidats fins a trobar-ne algun altre de vàlid (fins a trobar alguna solució compatible amb cadascun dels explorats):**
**   - ACTUALITZA < RE-MEMBER < MASCARA < FES-SUDOKU-1 (REAL = nil):**
**   - INI = T: 1, 2, 3, 4 i 5.**
**   - INI = nil: 1 i 2.**
** - Confirmació del candidat preestablert A-N i actualització:**
**   - ACTUALITZA < FES-SUDOKU-1 (REAL = T i INI = nil):**
**   - Si (assoc A-N TN/R-L) = T, no es repeteix l'execució d'ACTUALITZA-PLUS.**
**   - Si (assoc A-N TN/R-L) = nil: 1 i 2 (la inspecció 2 haurà estat supèrflua).**
- A FES-SUDOKU-2, quan es desactiva una casella registrada a FOTO2:
** - Exploració de tots els candidats (enregistrant totes les solucions compatibles amb cadascun d'ells):**
**   - ACTUALITZA < RE+MEMBER < ANORMALS < FES-SUDOKU-2 (REAL = INI = nil): 1 i 2.**

Atès que, a FES-SUDOKU-2, ACTUALITZA-PLUS < ACTUALITZA només aprofundeix fins els nivells 1 i 2, i per tal de justificar que no sigui imprescindible fer-ho fins els nivells 3, 4 i 5 per garantir temps d'espera raonables, tornarem a l'argumentació usada a l'annex precedent en relació al nivell 4: tot i que RE+MEMBER < ANORMALS ha d'escometre una exploració encara més feixuga que RE-MEMBER < MASCARA (no en té prou a mirar si hi ha alguna solució compatible amb cadascun dels valors candidats a la casella actual, sinó que haurà de trobar totes les solucions compatibles amb l'emplenament que queda en buidar aquesta casella), el grau d'ocupació serà força més elevat que el corresponent al moment d'activació de POST, tret que l'usuari s'entesti a multiplicar les desactivacions (per comparar-ho amb un altre moment crític de FES-SUDOKU-2), o a les situacions en què la detecció precoç de caselles sense candidats (detectant abans infraccions a la norma 2/3), de configuracions AAA-BBB o de conjunts tancats de caselles a efectes de valors admissibles (amb la depuració d'elements de R-LLL que encara mantenen informació obsoleta) resultava necessària per impedir que proliferessin les giragonses per un territori encara poc urbanitzat (a FES-SOLUCIO i FES-SUDOKU-1, en etapes incipients d'emplenament).

Així que ens limitarem a oferir el codi d'ACTUALITZA (que, en relació a la versió de referència, té la novetat d'incorporar la funció P1P2, introduïda en l'annex precedent i que, com recordareu, és compartida des de PLUS-N->P), RE-MEMBER (que, com passava en l'annex precedent, en la primera meitat INI = T ha de restringir al compliment de la condició (or (not 1-2) (= N GR)) l'activació de TN/R-L, que ara no només afectarà les depuracions desfermades a TERCETS i FILS/COLS sinó les més generals registrades en ACT-DEPURA < DEPURA), MASCA (que recull les modificacions que la versió de referència introduïa a l'antiga funció MASCARA) i RE+MEMBER (on l'unic canvi és redreçar la descurada expressió (foreach E L1A9 (if (member E R-N) ...) ...), substituint-la per (foreach E R-N (if E ...) ...), acció preconitzada al final del capítol de referència per a la millora formal de RE-MEMBER i MASCARA:

```
(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y XY L PLUS P-DEP L-DEP)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-P (list J K) A-N E
                        A-9*9 (ACT-9*9 A-N A-P A-9*9))
                  (P1P2))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P) FIL 0 COL 0)
             (if L-DEP
                 (setq A-LLL (ACT-DEPURA) DEP T L-DEP () XY ())
                 (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
                       A-LLL (ACT-LLL X Y A-LLL) XY ()))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P FIL COL)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if (or TT DEP)
                  (progn
                    (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)))
                    (if (or (not 1-2) (= N GR))
                        (setq 2R (list N (REORD R-LLL) (REORD REPLUS))
                              TN/R-L (if TT
                                         (subst 2R (cons N TT) TN/R-L)
                                         (if TN/R-L
                                             (cons 2R TN/R-L)
                                             (list 2R)))))))
              (if (> COL FIL)
                  (setq R-9*9 (TRANSPOSAR R-9*9)
                        R-LLL (TRANSPOSAR R-LLL) K ()
                        R-9*9 (foreach F (reverse R-9*9)
                                (setq J ())
                                (foreach E (reverse F)
                                  (setq J (cons (if (atom E) E (reverse E)) J)))
                                (setq K (cons J K)))))
              (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P))))))
  (if (not (or INI OK (setq OK (COMPLET-9*9 R-9*9))))
      (progn
        (if PRIMER (setq CAMI (cons R-P CAMI)))
        (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
```

```
        (foreach E R-N
          (if (and E (not OK) (not CANVI))
              (progn
                (if PRIMER
                    (progn
                      (setq CAMI (member R-P CAMI))
                      (if (and NOLLL (not (member CAUSANT CAMI)))
                          (setq CANVI T))))
                (if (not CANVI)
                    (progn
                      (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                      (if (not FORA)
                          (if (equal (cadr 2R) NIL*NIL)
                              (setq OK T)
                              (RE-MEMBER (car 2R) (cadr 2R)))))))))
  OK)

(defun MASCA (L POST / RR-9*9 RR-LLL PRIMER CAUSANT CAMI CANVI NOLLL RE-MEM)
  (setq LL () M 0 SS (ssadd))
  (foreach N L
    (if (and POST (not 1-2))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                 "el nombre...\""))
    (if (and N (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
             (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                 (progn
                   (setq RE-MEM (RE-MEMBER 0-**9*9** 0-LLL))
                   (if CANVI
                       (progn
                         (setq JJ (if (= (/ (cadr NOLLL) 3) 1)
                                      '(1 0 2)
                                      '(2 0 1))
                                  9**9 () I -1)
                         (foreach EE (F=3 RR-9*9 T)
                           (setq 0-L () I (1+ I))
                           (foreach E EE
                             (setq 0-L (cons (if (atom E) E (list (car E) I))
                                             0-L)))
                           (setq 9**9 (cons (reverse 0-L) 9**9)))
                         (setq RR-9*9 (reverse 9**9) RR-LLL (F=3 RR-LLL T)
                               PRIMER () CANVI ()
                               RE-MEM (RE-MEMBER RR-9*9 RR-LLL))))
                   RE-MEM)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-2)
              (progn
                (if POST (TREU-RETOL))
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
                (command "DESIGNA" (ssadd (entlast) SS) "")))
          (if (and POST (not 1-2)) (TREU-RETOL)))))

(defun RE+MEMBER (R-9*9 R-LLL / 2R R-N R-P F C0 C3 C6)
  (if (COMPLET-9*9 R-9*9)
      (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9) SUDOKUS-V*V)
            KKK (cons T KKK))
      (progn
        (if (and PRE (> (cadr R-P) 0))
            (setq PRE () F (nth 0 R-9*9)
```

```
                C0 (nth 0 F) C3 (nth 3 F) C6 (nth 6 F)
                NORM (and (< C0 C3 C6)
                          (< C0 (nth 1 F) (nth 2 F))
                          (< C3 (nth 4 F) (nth 5 F))
                          (< C6 (nth 7 F) (nth 8 F)))))
        (if (or PRE (not NORM))
            (progn
              (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
              (foreach E R-N
                (if E (progn
                        (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                        (RE+MEMBER (car 2R) (cadr 2R))))))
            (if NORM (setq PRE T))))))
```

Però havent parlat de redreçar una descurada expressió, a propòsit de RE+MEMBER, no podem deixar-ho aquí, perquè al llarg del programa també hi ha una descurada reiteració d'expressions, existint la funció SUPR. Això afectarà quatre funcions:
- K2- desapareixerà, en ser pràcticament igual que SUPR.
- A TERCETS substituirem els accessos (K2- N) i (K2- E/R-9) per (SUPR N K2) i (SUPR E/R-9 K2), respectivament.
- A FILS/COLS substituirem l'accés (K2- N) per (SUPR N K2).
- A FES-SUDOKU-2 substituirem l'assignació múltiple

```
    (setq I (assoc P S-V*V)
          J (reverse (cdr (member I (reverse S-V*V))))
          S-V*V (append J (cdr (member I S-V*V)))
          I (caar S-V*V) J (nth (car I) (nth (cadr I) **9*9**)))
  per
    (setq I (assoc P S-V*V)
          S-V*V (SUPR I S-V*V)
          I (caar S-V*V) J (nth (car I) (nth (cadr I) **9*9**)))
```

Igual com havíem acabat l'annex precedent destacant que el mètode 1 es confirmava com el més eficient, a conseqüència de les modificacions produïdes en ACTUALITZA-PLUS < ACTUALITZA, perquè pràcticament havíem eliminat l'eventualitat de llargues esperes en activar certes caselles, tot i que el preu a pagar era que la resposta a cada clic (l'espera mínima entre dos clics consecutius) es fes notar, ara hem de consignar el fet que els últims canvis en ACTUALITZA i RE-MEMBER aprofondeixen en aquesta tendència: d'una banda encara allunyem més el perill de llargues esperes, però el temps de resposta del primer clic (que va disminuint a mesura que augmenta el nombre de caselles activades) passa de dos a tres segons, circumstància que en opinió nostra no afecta la vigència del judici sobre la superioritat del mètode 1. De tota manera, cal relativitzar l'abast del mot "superioritat" i recordar que no contempla altre paràmetre que la velocitat del procés. En aquest mateix capítol, poc abans d'oferir la versió íntegra del codi, ja s'havia comentat que el mètode 1 no permetia controlar massa la forma (en actuar el dispositiu tapaforats, no només a efectes d'exploració sinó materialitzant els emplenaments automàtics) i deixant l'usuari a cegues en l'escaquer 9×9 (amb la tecla "<---Backspace" podia recular, però ningú no l'informava de l'encert dels emplenaments, en el sentit de saber si les caselles activades reduïen poc o molt el nombre de solucions compatibles), a diferència del mètode 2, en què l'única justificació de les dues llargues esperes (degudes a la voluntat de conèixer el nombre de solucions compatibles -de primer normalitzades i després totals-, en comptes de conformar-se a saber si n'hi havia una o més d'una) era poder anar informant l'usuari del progrés en la reducció del nombre de solucions compatibles, a fi de permetre-li rectificar en conseqüència.

Assegurar-se una visualització correcta

El funcionament de l'última versió resultava satisfactori, pel que fa al predomini visual dels valors escrits a la capa NORM-1 sobre els de la capa NORM-0 (esquerra) i dels escrits a la capa VAR-1 sobre els de la capa VAR-0 (dreta), amb l'ordinador de què l'autor disposava en el seu despatx de l'ETSEIB, on va desenvolupar gairebé tota la feina. Però, ja des del primer cop que es va endur el programa a l'aula on ensenyava AutoCAD, a les dependències de l'ETSETB, aprofitant els intervals entre classe i classe per provar-lo i anar-lo polint, va adonar-se que aquest predomini era força més precari.

De fet, la incorporació de la cautela **(command "BORRA" "LT" "" "UY")**, després de treure còpia del contingut d'una casella i d'haver-la transferit a la capa NORM-1 des de NORM-0 i/o a la capa VAR-1 des de VAR-0, en **PLUS-N->P** (funció subsidiària de **NC<2** i d'**ACTUALITZA-PLUS**) i **V+-** (funció subsidiària de **CLIC**), va ser un intent d'assegurar-se la jugada de cara a altres configuracions de sortida, perquè en els ordinadors de classe no tot funcionava tan bé: si es desestimava la drecera A-0 i s'optava per seguir els mètodes 1 o 2, l'activació de les caselles per convertir-les en públiques no sempre es reflectia visualment passant de gris a blanc/negre (b/n des d'ara), i la fallada tant podia ser que el contingut romangués en gris o que adoptés un aspecte híbrid (meitat en gris, meitat en b/n). Semblava com si no n'hi hagués prou a haver fet **SORTENTS** = **127** ni haver-ho rematat amb sengles ordres **(command "REGENT")** després dels **(command "CVPORT" 2)** i **(command "CVPORT" 3)** de la funció **CLIC**, fet que hauria d'haver garantit que els objectes es regeneressin en el mateix ordre en què havien estat dibuixats, i que una resposta tan imprevisible tingués més a veure amb les targetes gràfiques instal·lades en aquells ordinadors.

Amb l'adopció del nou dispositiu i l'eliminació dels **(command "REGENT")**, que més semblaven un obstacle que un ajut efectiu (altra cosa era usar **(command "REGEN")**, sense la **T** final, incorporat a la funció **V+-**, que almenys no entorpia la correcta visualització però que tampoc semblava indispensable), es va aconseguir que les coses anessin bé... mentre l'usuari mantingués una respectuosa distància amb la marxa del procés. Perquè si, cansat d'esperar passivament la conclusió d'un llarg procés de cerca, decidia aprofitar el temps treballant en paral·lel amb un altre programa, ¡ja havia begut oli!: en tractar de recuperar la finestra d'AutoCAD per veure com anava tot, situant-se per exemple sobre el cursor vertical que flanqueja l'àrea gràfica per la dreta, altre cop la preeminència dels nombres en b/n sobre els grisos se n'anava en orris. O si, fart de perdre el temps a l'expectativa d'un final que no arribava mai, decidia cancel·lar l'execució i tornar-la a rellançar però d'una altra manera i prenent nota abans de les caselles que havia arribat a activar, ¡que no se li acudís de moure aquest cursor o executar l'ordre **REGENT**!: altre cop el caos d'uns nombres en b/n i ara degradats a gris, del tot o a mitges. Però el risc de la desestructuració visual no es deturava aquí: si l'usuari havia estat pacient i havia fet bondat fins a la conclusió del procés, havent adoptat un mètode 2 en què no ens havíem molestat a fer cap depuració (la VARIANT ESCOLLIDA de la SUDOKU-SOLUCIÓ, normalitzada i sense normalitzar, restava íntegre en gris, amb sobreimpressió en b/n del SUDOKU-PROBLEMA) encara la podia esguerrar i perdre aquesta configuració de blancs, negres i grisos, si li donava per desencadenar una regeneració del dibuix des de l'Espai Paper o alterar-ne l'enquadrament.

Per fer front a tanta contumàcia ha calgut una solució dràstica: definir l'escena en 3D, situant el contingut de les capes blanques NORM-1 i VAR-1 a una cota **z** = **1**, per damunt dels continguts de les capes grises NORM-0 i VAR-0, i garantint que la percepció des del punt de vista d'un observador situat a $\mathbf{z} \rightarrow \infty$ sigui realista en tot moment mitjançant l'adopció de la modalitat **Oculto** de **MODOSOMBRA** (per defecte es funciona amb **estructura alámbrica 2D**, modalitat que mantindrem a l'Espai Paper, perquè és l'única que permetrà visualitzar la trama sòlida de color gris amb què omplim l'àrea rectangular compresa entre les finestres esquerra i dreta, servint de fons al caseller 3×3 amb els valors assignables, al símbol del Tao que gira amb el cursor quan volem convertir en pública una casella de la SUDOKU-SOLUCIÓ, o als textos que de tant en tant ens conviden a armar-nos de paciència i a esperar algun senyal per seguir introduint dades en alguna d'ambdues formes). Peró, com que la utilització d'aquesta modalitat és força més lenta i això es fa palès en dibuixar els 9×9 caràcters numèrics de cada finestra, preferim donar per bona la versió que hem ofert sencera en el capítol precedent i limitar-nos ara a indicar quins canvis

hauran d'introduir en el codi aquells usuaris que, en executar-la, s'hagin trobat que la resposta del seu equip presenta alguna de les anomalies descrites. Perquè sigui fàcil localitzar les diferències i realitzar les esmenes, reproduirem totes les funcions afectades (ometent-ne les parts més reiteratives, si la definició és llarga) per partida doble: de primer, a la versió original hi subratllarem el codi que hagi de desaparèixer; tot seguit, a la versió revisada hi subratllarem el codi incorporat de nou.

Per començar, en **PREPGRAF-9*9**, que tenia l'aspecte

```
(defun PREPGRAF-9*9 ()
  (setq G 251               ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
           "SAVETIME" 0
           .....................
           "PRESENTACION" "D" "SUDOKU"
           "SIMBSCP" "DES")
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvolgut/Benvolguda al programa SUDÒKULUM!  ")
  (prompt "*****************")
  (***) (***)
  (prompt "\n\n\nSUDÒKULUM no ha estat concebut per resoldre sudokus (tot i que ")
  (prompt "pot usar-se amb\naquest propòsit absurd, privant-nos del seu carácter")
  (prompt " lúdic) sinó per crear-ne.")
  ..............................................................................
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: aquella en què el ")
  (prompt "color GRIS es\n   diferencii bé del BLANC i el NEGRE, i doni bon ")
  (prompt "contrast entre text i fons.")
  (TRIA-G)
  (DIB-3*3 T)
  (command "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
           "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
           ..............................
           "SCP" "" "ESPACIOP"
           "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))
```

l'última funció **command** començarà la seva acció executant **MODOSOMBRA Oculto:**

```
(defun PREPGRAF-9*9 ()
  (setq G 251               ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
           "SAVETIME" 0
           .....................
           "PRESENTACION" "D" "SUDOKU"
           "SIMBSCP" "DES")
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvolgut/Benvolguda al programa SUDÒKULUM!  ")
  (prompt "*****************")
  (***) (***)
  (prompt "\n\n\nSUDÒKULUM no ha estat concebut per resoldre sudokus (tot i que ")
  (prompt "pot usar-se amb\naquest propòsit absurd, privant-nos del seu carácter")
  (prompt " lúdic) sinó per crear-ne.")
  ..............................................................................
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: aquella en què el ")
```

```
  (prompt "color GRIS es\n   diferencii bé del BLANC i el NEGRE, i doni bon ")
  (prompt "contrast entre text i fons.")
  (TRIA-G)
  (DIB-3*3 T)
  (command "MODOSOMBRA" "O"                              ; Només si és imprescindible.
           "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
           "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
           ..............................
           "SCP" "" "ESPACIOP"
           "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))
```

Pel que fa a **PLUS-N->P**, en lloc de les ordres **CAMBPROP**, **BORRA** i **REGEN**, executades mitjançant la penúltima funció **command**,

```
(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (if 1-2
          (progn
            (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
                  PPM (cons (list P1 P2 0) PPM)
                  LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                                LLL/LLL))
            (command "COPIA" "C" P1 P2 "" "0,0" ""
                     "CAMBPROP" "LT" "" "C" "VAR-1" ""
                     "BORRA" "LT" "" "UY" "REGEN"))
          (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N)))))
```

usarem l'ordre **CAMBIA** que, a diferència de **CAMBPROP**, inclou l'opció **eLevación**:

```
(defun PLUS-N->P (/ P1 P2)
  (setq A-N N A-P K PLUS T)
  (if REAL
      (if 1-2
          (progn
            (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
                  PPM (cons (list P1 P2 0) PPM)
                  LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                                LLL/LLL))
            (command "COPIA" "C" P1 P2 "" "0,0" ""
                     "CAMBIA" "LT" "" "P" "C" "VAR-1" "L" 1 ""))
          (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa N)))))
```

A **RASTRE** hi practicarem la mateixa substitució de **CAMBPROP** per **CAMBIA**,

```
(defun RASTRE ()
  (if 1-2
      (progn
        (if (= ABC "A")
            (command "CVPORT" 2 "BORRA" "C" "0,0" "8,8" "" "CVPORT" 3))
        (command "CAPA" "DES" "VAR-1" ""
                 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-1" ""
                 "CAPA" "ACT" "VAR-1" ""))
      (setq ABC "D"))
  (foreach E PPM (ssadd (ssname (ssget "_C" (car E) (cadr E)) 0) SS))
  (command "CAMBPROP" "C" "0,0" "8,8" "E" SS "" "C" (if 1-2 "VAR-0" "NORM-0") "")
  (if 1-2
      (progn
        (if (/= ABC "A")
            (command "CVPORT" 2 "CAPA" "DES" "NORM-1" ""
                     "BORRA" "C" "0,0" "8,8" ""
                     "CAPA" "ACT" "NORM-1" ""))
        (command "ESPACIOP" "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" ""))
      (command "ESPACIOP"))
  (RETOLS))
```

a fi i efecte de passar a cota **z** = **0** els caràcters seleccionats:

```
(defun RASTRE ()
  (if 1-2
      (progn
        (if (= ABC "A")
            (command "CVPORT" 2 "BORRA" "C" "0,0" "8,8" "" "CVPORT" 3))
        (command "CAPA" "DES" "VAR-1" ""
                 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-1" ""
                 "CAPA" "ACT" "VAR-1" ""))
      (setq ABC "D"))
  (foreach E PPM (ssadd (ssname (ssget "_C" (car E) (cadr E)) 0) SS))
  (command "CAMBIA" "C" "0,0" "8,8" "E" SS "" "P" "C" (if 1-2 "VAR-0" "NORM-0")
                                                       "L" 0 "")
  (if 1-2
      (progn
        (if (/= ABC "A")
            (command "CVPORT" 2 "CAPA" "DES" "NORM-1" ""
                     "BORRA" "C" "0,0" "8,8" ""
                     "CAPA" "ACT" "NORM-1" ""))
        (command "ESPACIOP" "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" ""))
      (command "ESPACIOP"))
  (RETOLS))
```

A **CANON->VARIANT** no hi filàvem massa prim, pel que fa a l'execució de **GRAF-9*9**, i procedíem a actualitzar els valors de les caselles d'ambdues finestres amb els de les pseudomatrius 9×9 ****V*V**** i ****V**V**** encara que no fos estrictament necessari. Però l'adopció del tipus de representació **MODOSOMBRA Oculto**, imprescindible per assegurar que el dispositiu **CAMBIA eLev 1** funcioni visualment, tot i que és més lent, ens obligarà a restringir l'accés a aquesta funció als casos indispensables: si l'usuari no vol aplicar a la SOLUCIÓ CREADA, COPIADA o CANÒNICA cap de les 10 permutacions que garanteixen que la VARIANT ESCOLLIDA seguirà sent una solució, no podrà evitar l'aparició d'aquest epítet inútil qualificant la finestra de la dreta però sí que li podem estalviar una espera absolutament prescindible quan ****V*V**** i ****V**V**** segueixen sent iguals.

```
(defun CANON->VARIANT (/ T1 T2 T3 T4 T5 T6 RESP OK)
  (graphscr)
  (setq T1 "\n   Assenyaleu (dreta) la " T2 " que vulgueu moure: "
        T3 "\n   Assenyaleu (dreta) on voleu situar-la: "
        T4 "\n   Assenyaleu (dreta) on voleu situar la " T5 " desplaçada: "
        T6 "\n   CLIC (centre) sobre el nombre " RESP "")
  (while RESP
         (if (/= (setq WWW (substr (getvar "LASTPROMPT") 1 35))
                 "Aquesta VARIANT ESCOLLIDA no permet")
             (prompt "\n."))
         (initget "F1 F2 F3 FS C1 C2 C3 CS TR PV")
         (setq RESP (getkword (strcat "\n.\nTipus de permutació: F1/F2/F3/FS/"
                                      "C1/C2/C3/CS/TR/PV <sortir>: ")))
         (if (not OK)
             (progn
               (if (= ABC "A")
                   (progn
                     (RETOL-1 '(-0.75 -0.55) "V. E. & SUDOKU INICIAL")
                     (setq **9*9** (FES-+9*9+-))))
               (setq OK T)
               (command "ESPACIOM")
               (if (= ABC "A") (GRAF-9*9 T))))
         (if (or (= RESP "F1") (= RESP "F3") (= RESP "C1") (= RESP "C3"))
             (F/C)
             (if (or (= RESP "F2") (= RESP "C2"))
                 (F//C)
                 (if (or (= RESP "FS") (= RESP "CS"))
                     (EXECUTA SIM)
                     (if (= RESP "TR")
                         (setq **9*9** (if (= ABC "A")
                                           (TRANSPOSAR **9*9**) **9*9**)
                               **9**9** (TRANSPOSAR **9**9**))
                         (if (= RESP "PV") (2-9V))))))
         (GRAF-9*9 (= ABC "A")))
```

```
  (initget "Si No")
  (setq CONF
        (getkword "\n.\n.\nUs sembla bé com a solució de treball (S/N)? <S>: ")
        CONF (if CONF CONF "Si")))
```

Però cal no oblidar que a **FES-SOLUCIO** havíem introduït una simplificació en el cas **(= ABC "B")**: en comptes de fer

```
            (command "DESPLAZA" CURSOR "" "0.1,0" ""
                     "ESPACIOM"
                     "CVPORT" 2 "CAPA" "D" "NORM-1" ""
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "CVPORT" 3 "CAPA" "D" "VAR-1" ""
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "ESPACIOP")
```

en cadascuna de les 9×9 iteracions, havíem fet

```
            (command "DESPLAZA" CURSOR "" "0.1,0" ""
                     "ESPACIOM"
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "ESPACIOP")
```

Això vol dir que, a diferència del cas **(= ABC "A")** en què ens limitàvem a omplir la finestra esquerra amb ****V*V**** i després copiàvem ****V**V**** = ****V*V**** a la dreta, o del cas **(= ABC "C")** en què l'emplenament de cada casella es feia en una i altra banda, la similitud amb aquest últim només era aparent: en realitat anàvem anotant els valors des de la finestra esquerra, com en el primer cas (en fer **ESPACIOM**, es passa directament a **CVPORT** = **2**), però ho fèiem sobre la capa PAPER, que no havia estat inutilitzada per **VGCAPA** des d'aquesta finestra (com VARIANT, VAR-1 i VAR-0) ni des de la finestra dreta (com NORMAL, NORM-1 i NORM-0); així que des d'ambdues finestres visualitzàvem els mateixos objectes. Per això, tot i haver decidit no aplicar cap permutació als elements de la SOLUCIÓ COPIADA, cal materialitzar allò que només era un pur joc d'aparences, donant-li accés a **GRAF-9*9**. En resum, com que aquest accés es fa massa feixuc amb **MODOSOMBRA Oculto**, quedarà restringit a dues situacions: que s'hagi aplicat alguna permutació a ****V*V**** (**RESP** no nul·la) o que es parteixi d'una SOLUCIÓ COPIADA (**(= ABC "B")**).

```
(defun CANON->VARIANT (/ T1 T2 T3 T4 T5 T6 RESP OK)
  (graphscr)
  (setq T1 "\n   Assenyaleu (dreta) la " T2 " que vulgueu moure: "
        T3 "\n   Assenyaleu (dreta) on voleu situar-la: "
        T4 "\n   Assenyaleu (dreta) on voleu situar la " T5 " desplaçada: "
        T6 "\n   CLIC (centre) sobre el nombre " RESP "")
  (while RESP
         (if (/= (setq WWW (substr (getvar "LASTPROMPT") 1 35))
                 "Aquesta VARIANT ESCOLLIDA no permet")
             (prompt "\n."))
         (initget "F1 F2 F3 FS C1 C2 C3 CS TR PV")
         (setq RESP (getkword (strcat "\n.\nTipus de permutació: F1/F2/F3/FS/"
                                      "C1/C2/C3/CS/TR/PV <sortir>: ")))
         (if (not OK)
             (progn
               (if (= ABC "A")
                   (progn
                     (RETOL-1 '(-0.75 -0.55) "V. E. & SUDOKU INICIAL")
                     (setq **9*9** (FES-+9*9+-))))
               (setq OK T)
               (command "ESPACIOM")
               (if (= ABC "A") (GRAF-9*9 T))))
         (if (or (= RESP "F1") (= RESP "F3") (= RESP "C1") (= RESP "C3"))
             (F/C)
             (if (or (= RESP "F2") (= RESP "C2"))
                 (F//C)
                 (if (or (= RESP "FS") (= RESP "CS"))
                     (EXECUTA SIM)
                     (if (= RESP "TR")
                         (setq **9*9** (if (= ABC "A")
                                           (TRANSPOSAR **9*9**) **9*9**)
                               **9**9** (TRANSPOSAR **9**9**))
                         (if (= RESP "PV") (2-9V))))))
         (if (or RESP (= ABC "B")) (GRAF-9*9 (= ABC "A"))))
```

```
  (initget "Si No")
  (setq CONF
        (getkword "\n.\n.\nUs sembla bé com a solució de treball (S/N)? <S>: ")
        CONF (if CONF CONF "Si")))
```

I a **V+-**, com hem fet a **PLUS-N->P**, en lloc de les ordres **CAMBPROP**, **BORRA** i **REGEN**,

```
(defun V+- (C)
  (if V
      (command "BORRA" (ssname SS 0) "")
      (command "COPIA" SS "" "0,0" ""
               "CAMBPROP" "LT" "" "C" C ""
               "BORRA" "LT" "" "UY" "REGEN")))
```

usarem l'ordre **CAMBIA** que inclou l'opció **eLevación**:

```
(defun V+- (C)
  (if V
      (command "BORRA" (ssname SS 0) "")
      (command "COPIA" SS "" "0,0" ""
               "CAMBIA" "LT" "" "P" "C" C "L" 1 "")))
```

Com que la tònica de les esmenes presentades és que les transferències des d'una capa gris a una b/n estiguin seguides d'un increment unitari en la coordenada **z**, convindria aclarir per què no ens hem molestat a modificar l'última línia de la funció **GRAF-9*9**

```
                  ......................
                  (if (and VAR (< I 0))
                      (command "CAMBPROP" "LT" "" "C" "NORM-1" "")))))))
```

amb l'afectació

```
                  ......................
                  (if (and VAR (< I 0))
                      (command "CAMBIA" "LT" "" "P" "C" "NORM-1" "L" 1 "")))))))
```

Efectivament, quan l'argument **VAR** d'aquesta funció és no nul (cosa que passa si partim d'una SOLUCIÓ CREADA, **(= ABC "A")**), si el valor de ****9*9**** és negatiu es transferirà des de la capa NORM-0 (gris) a la capa NORM-1 (b/n), com a segona part del dispositiu que havíem ideat per estendre la visualització del SUDOKU INICIAL a la fase "postsolució". La primera part d'aquest dispositiu era la reconversió de ****9*9****, marcant amb signe "-" els valors públics que constituïen aquest SUDOKU, i la duia a terme **FES-+9*9+-** en accedir a **CANON->VARIANT**. Només en aquesta última funció (que fa poc teníem ocasió d'examinar a propòsit de les restriccions posades a l'últim accés a **GRAF-9*9** aconsellades per la lentitud de **MODOSOMBRA Oculto**) s'hi donaran tals circumstàncies. Un cop abandonada **CANON->VARIANT**, la SOLUCIÓ CREADA (afectada o no per les permutacions proposades) es desdobla en la V. E. & SUDOKU INICIAL a l'esquerra i la VARIANT ESCOLLIDA (sense rastre de SUDOKU INICIAL) a la dreta, entrant de ple a la fase "postsolució" en haver d'escollir un mètode per arribar al SUDOKU-PROBLEMA: si triem 1, es manté visible la finestra de l'esquerra però només podrem actuar a la dreta, i només en ella es produiran superposicions que justifiquin el pas **z = 0 → z = 1**; si triem 2, la memòria visual del SUDOKU INICIAL es perd, la finestra esquerra és ocupada per la V. E. NORMALITZADA i, com que podrem actuar des d'ambdues finestres, tindrem superposicions a banda i banda. Anem per un camí o l'altre, el control passa a ser exclusiu de **PLUS-N->P**, **RASTRE** i **V+-**: vet aquí per què **GRAF-9*9** queda al marge del ball de coordenades **z**.

Una altra cosa que pot estranyar és per què, en arribar al final havent adoptat el mètode 2, a les caselles "públiques" el seu valor hi figura duplicat: en color b/n (a NORM-1 i VAR-1) sobre gris (a NORM-0 i VAR-0, respectivament). Perquè, de fet, si només hagués interessat assegurar la correcta visualització del resultat final, hi havia una alternativa al dispositiu **z = 0 → z = 1** que s'hauria reduït a una intervenció puntual: de la mateixa manera que en acabar-se l'execució ràpida (A-0) o l'execució ordinària pel mètode 1 (A-1, B-1 o C-1, tant li fa) la funció **RASTRE** assegurava que a cada casella només hi quedés un valor, en gris o en b/n, hauríem pogut realitzar una depuració final similar amb el mètode 2. Però ja hem comentat que també interessava preservar-la durant el procés d'anar fent públiques cada cop més caselles fins arribar a una única solució compatible, incloses algunes de les manipulacions a què l'usuari pogués sotmetre el sistema (accionar els cursors de desplaçament del dibuix, per exemple), i per aquesta raó s'havia considerat més eficient el dispositiu adoptat, resultant innecessari rematar el mètode 2 amb una

depuració final. Fins i tot podríem dir que millor no fer-la perquè, a diferència de l'aplicació del mètode abreujat (A-0) o del mètode 1, en què acabàvem tenint en pantalla dos escaquers, un amb el SUDOKU-PROBLEMA i l'altre amb la SUDOKU-SOLUCIÓ, que podíem imprimir conjuntament (després de desactivar les capes NORM-0 o VAR-0, si volíem), l'únic que el mètode 2 deixava en b/n era el SUDOKU-PROBLEMA, sobre la VARIANT ESCOLLIDA NORMALITZADA (esquerra) i sobre la VARIANT tal com raja (dreta): com que ambdues finestres eren operatives tota l'estona (a efectes d'activació de les caselles) i aquest va ser inicialment l'únic mètode (en el capítol precedent, *Un nou Camí i també una drecera*, l'autor comença explicant com en un moment donat va descobrir que allò que fins aleshores venia usant a la fase "presolució" podia utilitzar-se com a recurs alternatiu a la fase "postsolució", amb quatre canvis), en finalitzar el procés es va optar per deixar les coses tal i com havien quedat. Mantenint-ho així, la mandra que li fa a l'autor modificar aquest final per raons estrictament formals (que el resultat es mostri sempre de la mateixa manera, amb independència del mètode seguit) sempre es podria disculpar, considerant quant poc li costarà a l'usuari obtenir una presentació idèntica a la del mètode 1. Només li caldria fer les operacions següents:

- Activar la finestra esquerra i esborrar el contingut de l'escaquer (capes NORM-0 i NORM-1) fent **BORRA** amb una finestra **C** de **0,0** a **8,8**.
- Passar a la finestra dreta, desactivar la capa VAR-1 (eventualment fer **COPIA** amb una finestra **C** de **0,0** a **8,8**, i desplaçament **0,0**, si es vol mantenir un exemplar de la VARIANT ESCOLLIDA a la capa VAR-0) i transferir amb CAMBPROP el contingut d'una finestra **C** de **0,0** a **8,8** (havent fet **COPIA** la selecció pot ser amb **P**revio) des de la capa VAR-0 a la NORM-1, acabant amb la reactivació de la capa VAR-1.

Tanmateix, aquest canvi esdevindria força més complicat si el contingut de la capa VAR-0 no fos una matriu completa de 9×9 caràcters numèrics sinó que presentés uns buits just on la capa VAR-1 els mostra plens. Ara bé: com que tota iniciativa per millorar o superar SUDÒKULUM serà ben rebuda (no per altra cosa se'n publica el codi font, embolcallat en un text que voldria ser més explicatiu que soporífer), el lector perfeccionista que vulgui agafar el relleu i prendre's la molestia de modificar SUDÒKULUM per assolir l'homologació suggerida és ben lliure de fer-ho.

En relació al procediment de captura de resultats que a cada usuari li pugui ser més còmode (des del guardador sistemàtic al episòdic), i ja com a postil·la final, un advertiment més: si trobeu que els noms de funcions i variables no són massa inspirats (uns massa curts i desprovistos de connotació, i d'altres que semblen tenir-ne però amb uns referents significatius que no venen al cas), probablement tindreu raó. Passa que el programa ha experimentat tantes metamorfosis sobre la marxa que, en arribar a la versió definitiva, moltes de les denominacions ja no tenien massa a veure amb el rol efectivament assignat. Però hi ha quatre d'aquests noms absurds que, encertadament o no, han estat pensats amb abundància d'asteriscs amb tota intenció. Parlem de les variables ****9*9****, ****9**9****, ****V*V**** i ****V**V****, i el motiu d'aquestes denominacions tan rocambolesques ha estat la previsió d'una hipòtesi prou versemblant: que l'usuari decidís excloure-les de la declaració de variables de **C:SUDOKULUM** i **FES-SUDOKU** per tal de tenir-hi accés un cop executat el programa (com a variables globals en la sessió de dibuix), disposant-ne lliurement per capturar els valors de les caselles del SUDOKU-PROBLEMA i la SUDOKU-SOLUCIÓ, en el cas que no n'hi hagués prou a imprimir el dibuix; i en aquest supòsit calia noms d'improbable coincidència accidental amb els d'altres variables globals.

****
****
****

Actualització a l'última versió de FES-SOLUCIO del capítol *6- Etapa prèvia: crear una solució, IV (... o a Camprodon)*.

Tot i que podíem donar-ho per suposat, en haver passat part de la funció PLUS-N->P a P1P2, deixarem clar que és la segona funció la que hereta el conflicte visual i, atès que el codi subratllat no donava bon resultat amb altres targetes gràfiques,

```
(defun P1P2 (/ P1 P2)
  (if 1-2
      (progn
        (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
              PPM (cons (list P1 P2 0) PPM)
              LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                            LLL/LLL))
```

```
        (command "COPIA" "C" P1 P2 "" "0,0" ""
                 "CAMBPROP" "LT" "" "C" "VAR-1" ""
                 "BORRA" "LT" "" "UY" "REGEN"))
      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa A-N))))
```

hi hem hagut de practicar la substitució que abans aplicàvem a la primera, fent

```
(defun P1P2 (/ P1 P2)
  (if 1-2
      (progn
        (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
              PPM (cons (list P1 P2 0) PPM)
              LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                            LLL/LLL))
        (command "COPIA" "C" P1 P2 "" "0,0" ""
                 "CAMBIA" "LT" "" "P" "C" "VAR-1" "L" 1 ""))
      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa A-N))))
```

I amb això hauríem enllestit el present capítol, si no fos perquè la tautològica però implacable Llei de Murphy sempre s'acaba complint: en aquest cas, l'amistosa invitació a fotre el camp d'una vegada (del despatx de l'E.T.S.E.I.B.), anunciada ja en la segona pàgina de la *Introducció*, que es va produir pel febrer de 2011 i que va venir d'on menys s'ho hagués esperat l'autor. Que no fos cap buròcrata sinó un professor, company de Departament a l'E.T.S.E.I.B., potser el que més admirava malgrat el poc tracte, per haver acceptat la responsabilitat de ser Cap de Secció en un moment delicat (en dos anys s'havien jubilat quatre professors veterans de la plantilla i s'havia quedat ell, la Sra. L de qui hem parlat en la *Introducció* i un altre professor a temps complet, havent de convocar a corre-cuita dues places d'auxiliar a temps parcial per poder fer front a les matèries reglades) i plantar cara al sector que (dedicat a la "transferència de tecnologia" i altres empreses de major reconeixement institucional) pretenia l'hegemonia i desentendre's de les obligacions menys agraïdes, i que ho fes justament apel·lant a la conveniència de no donar més munició a aquest sector (*vosaltres, tan estrictes, ¿per què consentiu que un jubilat segueixi freqüentant les dependències departamentals?*), va agafar-lo desprevingut i no s'hi va poder negar. Només va tenir esma d'objectar allò que també recordareu de la segona pàgina de la *Introducció*: la necessitat de seguir amb el mateix ordinador (el més senzillet dels 5 que hi havia en l'antic despatx), perquè el treball s'havia realitzat amb la versió ACAD 2004 i perquè després del fotimer d'hores invertides no volia arriscar-se a traslladar-s'ho tot a un altre ordinador i quedar penjat o haver de passar-se a una versió 2011 (l'única que en aquell moment podia instal·lar en l'ordinador que tenia a casa) que, sens dubte, s'hauria transformat força i l'obligaria a revisar moltes coses des del principi. Ell mateix (l'autor) es va ficar a la ratera perquè, com que realment el destí del seu equip hauria estat la propera tramesa de material informàtic obsolet per al Tercer Món (el monitor, l'unic CRT que quedava, ni això, sinó que hauria anat de dret a la deixalleria), l'astut Cap de Secció va saltar amb un *Tots els problemes fossin com aquest!*, i va anunciar-li un regal (enverinat): faria desinventariar ordinador i monitor i se'ls podria endur a casa. Això li va caure com una dutxa d'aigua freda, perquè (per què ens hem d'enganyar?) la possibilitat d'anar-se'n de casa unes hores al dia i seguir immers en un ambient de treball (a nivell purament *voyeur*, perquè fins i tot quan impartia docència en l'E.T.S.E.T.B. no hi havia cap integració real amb l'equip departament de l'E.T.S.E.I.B.), era l'altre motiu real del seu interès a seguir anant diàriament al Campus Sud, però va haver de dibuixar un ampli somriure de satisfacció i expressar l'agraïment perquè l'haguessin salvat la vida. Deu dies després ja tenia a casa la carraca amb la versió 16.0 d'AutoCAD (2004) i el portàtil on una setmana abans havia instal·lat la versió 18.1 (2011).

Aquesta història, continuació de l'explicada en la *Introducció*, ve a tomb perquè, com era d'esperar, el trasllat va afectar considerablement la marxa del treball: més enllà dels problemes psicològics d'adaptació a la rutina diària lligada a un lloc concret, quedava l'adaptació al nou ordinador i a la nova versió d'AutoCAD. Per que es pugui entendre, l'autor no tindrà més remei que revelar una part de la seva personalitat i utilitzar un castellanisme, *papanatisme*, en no haver reeixit a trobar algun mot català amb les mateixes connotacions que el *papanatismo* castellà: *Actitud que consiste en admirar algo o a alguien de manera excesiva, simple y poco crítica*. Tot això, per començar afirmant que una de les formes més virulentes que coneix de consumisme compulsiu és el que podríem anomenar papanatisme tecnològic. L'autor no és cap neoludita, sinó de l'opinió que qualsevol innovació tecnològica

és positiva, en el sentit de ser potencialment beneficiosa per a algú; una altra cosa és per a qui o en quines condicions serà profitosa. Per exemple, reconeix que la telefonia mòbil haurà canviat la vida (en el bon sentit) a molts professionals, fins i tot a gent amb una vida pública intensa, o amb força família o amics. Però, com que cap d'aquests supòsits respon al seu perfil personal, l'autor no té mòbil. El mateix passa amb els ordinadors. Tret de l'*Apple-II Europlus* adquirit en 1982 i compartit 4 anys amb els senyors J i M (vicissituds de la seva tesi doctoral, a la *Introducció*), no es va comprar cap ordinador fins l'any 2010, i només ho va fer en previsió del que ja es veia venir i que finalment es va produir en febrer de 2011. Però, ¿que no treballava amb ordinadors, i fins i tot impartia classes d'AutoCAD?: es clar que sí, però aquesta era la seva feina a la U.P.C. i la feia utilitzant el material informàtic que aquesta posava al seu abast. És clar que tampoc us pretén explicar una pel·lícula d'indis: si hagués tingut fills hi hauria hagut ordinadors a casa, però com que aquest no era el cas i les vuit hores diàries a la U.P.C. ja li donaven per a tot, incloent-hi la gestió del correu electrònic, no n'hi calia.

D'altra banda, les condicions de treball a casa no eren òptimes, sobretot des que l'antiga dependència amb dues taules de dibuix s'havia convertit en una saleta amb un armari mastodòntic i la de reunions en quarto dels mals endreços. Va limitar-se a buidar una mica aquest traster virtual, per encabir-hi l'antic ordinador, el nou i una petita impressora. Però encara quedava algun tema pendent, perquè tothom té les seves manies i hi havia dues coses a què el Sr. K de la *Introducció* es negava:

- No era només qüestió d'espai físic sinó que, com ja s'ha dit, no es veia reclòs a casa tot el dia; trenta anys enrera ho havia experimentat, intentant repartir espai i temps entre vida domèstica i treball d'arquitecte, i va tenir problemes.
- Teòricament, podia desenvolupar el que tenia entre mans sense connectar-se a La Xarxa, però això implicava renunciar a l'ajut *on line* amb ACAD 2011 (amb 2004 no li calia); a la consulta (també *on line*) del Diccionari de la Gran Enciclopèdia Catalana (l'únic que li facilitava l'accés a la conjugació completa dels verbs); a poder-se distreure una mica amb la Viquipèdia o Google Maps, quan s'embussava amb SUDÒKULUM o amb la redacció del text explicatiu; i, el més important, a usar el correu electrònic. ¿Hi havia algun problema per accedir a Internet? Només un: que no li donava la gana de pagar peatge a cap operadora de telecomunicacions.

Ambdues coses les podia resoldre anant a treballar a biblioteques públiques o de la U.P.C., però a aquests llocs no podia endur-se l'ordinador antic, sota el braç. Era en previsió d'això (per poder-se'l endur sota el braç, o gairebé) que s'havia comprat un portàtil amb pantalla de 17,3", format fins i tot un pèl més gran que el monitor CRT (per bé que s'hagués estimat més que no fos panoràmica), donant per fet que no li resultaria difícil instal·lar-hi una versió 16 d'AutoCAD, però, ves per on, amb això es va equivocar. Com haureu llegit a la *Introducció*, a diferència del seu excompany el Sr. J, al Sr. K més aviat el treia de polleguera que cada any Autodesk canviés les regles del joc en un percentatge considerable, amb el pretext d'innovacions (com aquell acudit una mica beneitó, de fa uns cinquanta anys, que posava en boca d'un avi: "Ara que havia après a dir *penícula*, ara en diuen *flim*"), i no estava disposat a perdre massa temps a estudiar l'ús de la versió 18 per tal de salvar els esculls que indubtablement sorgirien en rodar-hi SUDÒKULUM, tal com el tenia: com que els problemes es presentarien en l'àmbit visual, confiava que no arribessin a fer impossible les entrades i sortides gràfiques, de manera que se'n podria sortir treballant a casa només quan calgués verificar la visualització (amb el vell ordinador fix i ACAD 2004) i fent-ho a fora (amb el portàtil i ACAD 2011) quan calgués idear, implementar, provar (mitjançant expressions provisionals **princ** o **prompt**, variables externes o fins i tot sortides gràfiques defectuoses, mentre fossin intel·ligibles) i optimitzar processos interns, que seria el més freqüent.

Però això que en teoria semblava tan senzill es va anar complicant: la Biblioteca de l'E.U.E.T.I.B. (que és on ha estat anant habitualment) i la Biblioteca Agustí Centelles (que és on anava els dissabtes i, en general, quan la primera no obria), ambdues a Barcelona, i la Biblioteca Pública de València (quan s'estava en aquesta ciutat) han estat testimonis de les ventures i desventures d'aquest pobre pecador. Ell, que mai havia estat una rata de biblioteca, se'n va convertir en assidu i, és clar, com que d'entrada havia hagut de fer uns mínims ajustos per fer rutllar ACAD 2011, cada vegada li feia més mandra comprovar a casa si la traducció gràfica dels progressos assolits era correcta, cosa que l'obligava a portar en paral·lel dues versions de SUDÒKULUM. Així que, quasi sense voler, va anar arranjant tots aquells aspectes del programa que inicialment no tenien un tractament visual satisfactori en ACAD 2011, fins a tenir dues versions plenament operatives, per a ACAD 2004 i ACAD 2011, i va pensar que, havent-se pres la molèstia, seria bo donar a conèixer les dues amb el benentès que, programant per a la 1ª, oferíem adaptacions a la 2ª.

Però abans de presentar aquestes adaptacions, i aprofitant que per introduir-les hem proseguit la narració d'una història que vam començar en aquella *Introducció sense vaselina*, l'autor vol fer un parèntesi per contextualitzar algunes de les rotundes opinions que hi va deixar anar. En primer lloc per reconèixer que arriba tard a tot arreu: ¡mira que confessar d'haver descobert que la socialdemocràcia era l'única esquerra transformadora possible, quan ja és morta i només queden uns partits centristes que són ovelles disfressades amb pell de llop (de fet, fa prop d'un segle ja se'ls va veure el llautó, als partits socialdemòcrates europeus)! Entre altres coses, a Catalunya i a Espanya els governs socialdemòcrates passaran a la història per no haver aprofitat l'oportunitat d'escometre una regeneració més seriosa de la Funció Pública, i amb això se li ha servit a la dreta que és ara en el poder, en safata de plata, la coartada perfecta per procedir sistemàticament a delmar-la i acabar-la de desvirtuar: només faltava la crisi econòmica i el mite neoliberal que la privatització dels serveis públics n'assegura qualitat i òptima administració, àmpliament desmentit per l'experiència on l'han posat en pràctica.

Perquè, admetent que competència i afany de lucre siguin empremptes evolutives més antigues i marcades que la tendència a la cooperació i, en conseqüència, el mercat sigui el millor instrument per a la creació de riquesa, l'Estat ha de regular-lo (per assegurar-ne una distribució més correcta) i, eventualment, intervenir-hi o substituir-lo (per garantir serveis no rendibles però necessaris), cosa que només pot donar-se si el poder polític controla el poder econòmic i no a l'inrevés. Des d'aquesta perspectiva, l'Estat és un mal menor: no pot confondre´s amb la societat civil ni suplantar-la, però ha d'existir; ser fort, però sota control democràtic (això, i no pas fer apologia del liberalisme, volia dir l'autor quan al final de la pàgina 4 escrivia: "UNA SOCIETAT DE FUNCIONARIS NO FUNCIONARÀ MAI A LA VIDA!").

La reducció dels anomenats dies *moscosos*, *canosos* i altres bicoques funcionarials de difícil homologació europea només farà que donar més ales a la picaresca, si va acompanyada de retallades retributives. És imperatiu dur la flexibilització dels horaris fins on sigui tècnicament possible, per a una major conciliació de la vida familiar i laboral, però caldrà conscienciar al personal de la Universitat Pública (no només al P.A.S. sinó al P.D.I.), ni que sigui apel·lant a la seva condició de contribuent, que el compliment de les hores establertes no sols ha de ser estricte sinó verificable, si es vol començar a dissipar el secular recel de la ciutadania. Quant a la privatització encoberta, en marxa des de fa molts anys, el retrocés del finançament públic no farà sinó accelerar-la. La poc honesta compatibilització de l'exercici particular d'una professió amb la dedicació docent a temps complet (amb l'adquisició d'una butlla per el mòdic 14,7% dels ingressos corresponents) s'anirà extingint probablement, "gràcies" a l'actual programació de la carrera docent que, concebuda com a carrera d'obstacles, tampoc no garanteix la captació dels millors. Pel que fa a aquests dubtosos emprenedors sota aixopluc de la Universitat, sense la clàssica legitimació del risc però que també cobren per partida doble, hem de suposar que proliferaran al mateix ritme que ho facin els Patronats, Fundacions i Càtedres d'Empresa. Com li feia proclamar Voltaire al seu personatge Pangloss, tot parodiant Leibniz, "tot succeeix a fi de bé, i en el millor dels mons possibles".

L'autor hauria deixat aquí aquestes opinions, però el Déu dels soviets (com solia dir l'enyorat Manuel Vázquez Montalbán) haurà volgut que el 20 de març de 2013, quan li quedava menys d'una setmana per acabar aquest document, rebés l'e-mail que reprodueix textualment (el subratllat és de l'autor), tot sentint vergonya aliena:

> *Benvolguts companys, benvolgudes companyes,*
>
> *Atenent l'acord del Consell de Govern del passat dia 15, us comunico que l'ETSEIB romandrà tancada durant la Setmana Santa i, per tant, no hi hauran els serveis habituals.*
> *Tot i això, es facilitarà l'accés a totes les persones de l'Escola que necessitin desenvolupar-hi activitats imprescindibles, sent suficient la seva identificació per contrastar-la amb les dades del cens. Es necessitarà tramitar autorització específica per a totes aquelles persones que no constin al cens. En el cas de l'estudiantat la identificació serà amb el carnet universitari.*
>
> *L'horari de possible accés i permanència a l'Escola serà:*
> *- els dies 25, 26 i 27, de 7:30 a 21 hores*
> *- el dia 28, de 7:30 a 15 hores.*

S'ha de tenir en compte que durant els dies 25, 26 i 27 es produiran petites interrupcions programades del subministrament elèctric per revisions obligatòries del sistema, tal com s'ha comunicat des del Servei de Manteniment.

Agraeixo la vostra col·laboració.

Atentament,

Francesc Roure
Director ETSEIB

Els acords adoptats en el Consell de Govern de la U.P.C. i sancionats pel Consell Social quatre dies més tard han tingut un ampli ressò a la premsa d'aquests dies (l'autor us pot citar l'article "La Generalitat intervé en el pla de sanejament de la Politècnica", a la p. 27 d'EL PERIÓDICO DE CATALUNYA del 20/03/2013), malgrat la saturació de notícies de primera plana (nacionals, estatals i internacionals), però l'esmentat en l'*e-mail* no hi figurava, atesa la insignificància periodística del fet en relació a d'altres de més transcendents (110 milions d'euros de dèficit acumulat per la U.P.C.; acomiadament o no renovació de contracte de 250 P.D.I. i 90 P.A.S...). Però té molt a veure amb allò que comentàvem el el primer paràgraf de la *Introducció* i en el penúltim d'aquí: "... caldrà conscienciar al personal de la Universitat Pública (no només al P.A.S. sinó al P.D.I.), ni que sigui apel·lant a la seva condició de contribuent, que el compliment de les hores establertes no sols ha de ser estricte sinó verificable, si es vol començar a dissipar el secular recel de la ciutadania". En comptes d'això, ja podeu veure per on van els trets: enguany ja no passarà només que, de dilluns a dijous al migdia (el 25, 26 i 27, i matí del 28, que són dies feiners), no hi hagi pràcticament ningú a l'E.T.S.E.I.B. sinó que una vergonyosa realitat, tolerada per l'autoritat acadèmica, assoleix el reconeixement oficial i fins i tot és recomanada, tret que es tracti d'activitats imprescindibles. Sembla, doncs, que només els 11 membres del Consell de Govern que van votar en contra de l'acord "Tancament de dilluns a dijous de Setmana Santa", creien que complir amb les obligacions laborals (que, per més inri, estan pagades amb els impostos d'una majoria de contribuents que tenen controlat escrupolosament el seu temps de treball) és una activitat imprescindible; la resta, o no ho tenia clar (13 vots en blanc) o creia que no (16 vots a favor).

Després d'aquest segon i últim esbravament, l'autor enumerarà les característiques del ordinador DELL Vostro 3700 que va comprar-se; més que res, perquè amb aquesta informació el lector interessat a utilitzar SUDÒKULUM, rodant-lo sobre ACAD 2011, tindrà una referència orientativa per poder adaptar el codi al seu equip, cas que la visualització no resultés del tot satisfactòria, perquè no trigarà a veure que el mètode seguit aquí ha estat excusivament empíric, en no tenir l'autor cap mena d'experiència práctica en l'ús d'aquesta versió d'AutoCAD (ni el menor interès en adquirir-ne), i podria ser que algunes mesures adoptades no fossin necessàries amb un altre *hardware*:
- Processador: I5-430M (2,26 GHz).
- Memòria: 4 GB (2 × 2.048 MB), DDR3 a 1.333 MHz.
- Targeta gràfica: Nvidia Geforce GT330M de 1 GB.
- Pantalla: 17,3" panoràmica LED antireflectant, d'alta definició (1.600 × 900).
- Disc dur: SATA a 7.200 rpm, de 320 GB.

Sistema operatiu instal·lat: Windows 7 Home Premium, de 32 Bits.

Abans de passar a les correccions necessàries per mantenir amb el nou equip les condicions de visualització de l'antic, l'autor us farà un advertiment que també és una nova confessió. Com podeu suposar, el microprocessador del portàtil és més ràpid que el del vell Pèntium apedaçat que va dur a casa i, com que pràcticament tota la feina realitzada des de febrer de 2011 s'ha desenvolupat amb el portàtil, hi havia un petit problema a resoldre: homologar totes les referències a la durada dels diferents processos. Igual com s'ha decidit que la versió de referència fos (seguís sent) ACAD 2004, els temps de referència correspondran als de l'ordinador fix. Però això no vol dir que tots els que sumàriament són posteriors al capítol 11 siguin de primera mà: en alguna ocasió l'autor s'ha pres la molèstia de repetir en el fix un procés madurat i mesurat en el portàtil, però la major part de les xifres que figuren a partir del capítol 12, sobretot si són elevades, han sortit d'aplicar un procediment tan expeditiu com poc rigorós: multiplicar per 1,35 (que era la mitjana de les ràtios entre durades, centrades bàsicament en les ràfegues informatives de **COMPAT-PERFILS**) totes les lectures realitzades amb el portàtil.

Davant del primer contratemps que es va presentar, l'autor va ser expeditiu, i no se'n penedeix perquè, encara que s'hagués entestatat a treure'n l'entrellat i se n'hagués sortit, de segur que hauria acabat fent el mateix, que ara us aclarirem. No es tracta únicament d'un problema que incideix de ple en el títol del capítol, *Assegurar-se una visualització correcta*, sinó que fa referència a la modificació bàsica que, sempre amb la mirada posada en ACAD 2004, havíem adoptat no fa massa: situar en una elevació **z = 1** els objectes que, encavalcats amb d'altres, volíem que predominessin visualment. Per això cal anar a l'inici del capítol (pàg. 388).

No només recordarem algunes coses que havíem dit allà sinó que detallarem algunes circumstàncies que havíem obviat, perquè sense aquestes precisions és difícil de justificar el que farem aquí. El problema de predominis visuals no desitjats entre objectes superposats el teníem entre els continguts de les caselles (capa NORM-0 i NORM-1, a l'esquerra; capa VAR-0 i VAR-1, a la dreta) i no entre els textos i el fons gris de la franja entre finestres. Per això, l'elevació **z = 1** dels caràcters numèrics situats a les capes NORM-1 i VAR-1 i l'adopció de la modalitat **Oculto** els practicàvem a l'Espai Model. En l'Espai Paper ja ens funcionava tot bé: suposant que haguéssim pogut adoptar també la modalitat **Oculto** (que no és el cas) tampoc no ens hauria convingut de fer-ho, perquè els tramats sòlids no es visualitzen i això hagués comportat no poder usar-los per crear la franja gris entre les finestres. Les mostres les dibuixàvem a l'Espai Model (també les hauríem pogut fer a l'Espai Paper, però hagués estat una complicació innecessària) perquè en els requadres 3×3 situats a l'esquerra de les mostres (fabricats per **DIB-3*3**, funció que tot seguit aprofitàvem per a l'escaquer 9×9) no hi havia superposició de valors. I havíem de fer-les en modalitat **estructura alámbrica 2D** per visualitzar el tramat sòlid gris. Així s'explica que, en **PREPGRAF-9*9**, haguem fet **(command "MODOSOMBRA" "2D")** abans dels textos de benvinguda al programa i de la presentació i elecció de les mostres en **TRIA-G**, i que haguem esperat a la conclusió d'aquests preliminars per executar **(command "MODOSOMBRA" "O")**. (Si la fabricació del requadre 3×3 de base amb **DIB-3*3** la fèiem abans sols era per no interferir la llarga sèrie d'arguments de **command**.)

Però amb ACAD 2011 (versió en què, per cert, cal triar entre **-MODOSOMBRA Oculto** o **MODOSOMBRA ocUlto**) això ja no és possible: per raons ignotes (i que l'autor tampoc no s'ha sentit gens motivat per escatir), si interrompem el programa just després de **(command "-MODOSOMBRA" "O")** i cursem l'ordre **CAMBPROP** interactivament (fins i tot si escrivim en temps real l'expressió **(command "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G "")** i l'executem), les quatre línies interiors del requadre (2 horitzontals i 2 verticals) resultaran seleccionades, però si això ho fem en diferit, executant **C:SUDOKULUM** després d'haver avaluat aquesta i les altres funcions de l'arxiu .LSP, la captura serà infructuosa (com podrem copsar posant fora de servei l'expressió **(setvar "CMDECHO" 0)**, anterior en **PREPGRAF-9*9**, pel missatge *0 encontrados*), amb la previsible alteració en la interpretació dels subsegüents arguments de **command**. I no és pas que el canvi a la segona modalitat de representació (o estil visual, com ara en diuen els manuals) no s'hagi produït, perquè una mena d'aclariment del fons negre (que deu ser l'opció per defecte de **OPCIONES > Visual. > [Elementos de ventana]: Colores > Contexto: Proyección paralela 3D -> Elementos de interfaz; Fondo uniforme -> Color: 33,40,48**) n'és la millor confirmació, tant si hem fet **(command "-MODOSOMBRA" "O")**, **(command "MODOSOMBRA" "U")**, **(command "MODOSOMBRA" "OCULTO")** o **(command "ESTVISACTUAL" "U")**. Si alguna cosa resulta evident és que les implicacions d'aquests estils visuals són força més complexes en ACAD 2011.

Hi hauria una manera de prescindir de la segona intervenció de l'ordre **MODOSOMBRA**: que la primera marqués ja el canvi a la modalitat **ocUlto**, però llavors hauríem de substituir el tramat sòlid dels rectangles de mostra, tenyits de diferents grisos, per un tramat convencional de línies paral·leles (no predefinit, sinó definit per l'usuari) i prou atapeït com per aconseguir un efecte de continuïtat en el color. Però no val la pena molestar-s'hi, perquè tampoc no serviria de res: no únicament seguirien produint-se problemes a l'Espai Model (en la visualització de caselles), sinó que ara en sortirien de nous a l'Espai Paper (en la visualització dels rètols i en la del fons gris). La raó és que, des de la versió 16.0 d'AutoCAD (ACAD 2004) fins a la 18.1 (ACAD 2011), el programa s'ha anat dotant de recursos (ordres i/o varibles de sistema) específics per a l'ordenació d'objectes encavalcats, de cara a la seva visualització, i només adoptant el nou instrumental podrem controlar-la.

Fet i fet, per adaptar-nos a ACAD 2011 és més pràctic ometre la segona intervenció de **MODOSOMBRA**, deixar la primera com estava i adoptar els recursos específics de la versió que, com veureu, tampoc no són tants com per haver de segregar fragments importants de codi. Com que el present treball segueix estant concebut bàsicament

per a ACAD 2004, i només les circumstàncies que abans hem referit justificaven un mínim esforç per poder seguir rutllant amb ACAD 2011 (esforç que ha semblat oportú d'amortitzar incorporant-hi les adaptacions com a variants, per tal de beneficiar als lectors que disposin d'aquesta versió o de versions més pròximes), l'autor ha preferit no integrar en el codi les variants *ad hoc* mitjançant condicions com ara

```
(if (or (= (substr (getvar "ACADVER") 1 2) "15")
        (= (substr (getvar "ACADVER") 1 4) "16.0"))
    (...)
    (...))
```

o bé

```
(if (= (substr (getvar "ACADVER") 1 4) "18.1")
    (...)
    (...))
```

perquè amb això s'admetrien a tràmit terceres versions probablement inadaptades al codi, ni tampoc amb condicions tancades com ara

```
(if (or (= (substr (getvar "ACADVER") 1 2) "15")
        (= (substr (getvar "ACADVER") 1 4) "16.0"))
    (...)
    (if (= (substr (getvar "ACADVER") 1 4) "18.1")
        (...)
        (progn (AVIS-DE-VIA-MORTA) (exit))))
```

perquè, tot i que posades al començament no comportarien molèsties ni sorpreses, seria embolicar la troca encara més que les condicions anteriors i, en definitiva, significaria posar les tres versions comprovades en la mateixa línia de sortida. Sembla més adient (i més fàcilment manipulable per l'usuari no expert) la fòrmula de neutralitzar les línies de codi destinades a una versió diferent d'ACAD 2004 amb un caràcter ";" a l'inici de línia, amb un comentari a la dreta (precedit del seu ";", per a quan l'altre s'esborri), com ho havíem fet amb el codi alternatiu dedicat a ACAD 2000. Pot estranyar que únicament una línia de **PREPGRAF-9*9** dugui a la dreta el comentari "; Només si és imprescindible.": **"MODOSOMBRA" "O"**, que és la primera ordre executada mitjançant l'ultima funció **command** (més endavant, quan a **PREPGRAF-9*9** haguem incorporat les variants corresponents a ACAD 2011, la partirem en dues línies, perquè cap distracció no pugui dur a esborrar també **"(command "**).

El lector pot preguntar-se, molt raonablement, per què no hem afegit aquest mateix comentari a la dreta de totes les línies en què **CAMBPROP** era substituït per **CAMBIA Propiedades**, per canviar l'elevació dels objectes seleccionats amb **eLev 1** (**eLev 0**, en el cas de la funció **RASTRE**). I és que, al final d'aquella pàgina 469, quan vam afirmar que "la utilització d'aquesta modalitat [referint-nos a **MODOSOMBRA Oculta**] és força més lenta i això es fa palès en dibuixar els 9×9 caràcters numèrics de cada finestra", s'hauria d'haver especificat que allò que ralentitzava el dibuix de l'escaquer no era pas haver posat els caràcters numèrics en blanc (de les capes NORM-1 i VAR-1) a una elevació **z = 1** sinó visualitzar-los amb la modalitat **Oculta**. Així doncs, posats a deixar-li les coses fàcils a l'usuari, no caldrà esborrar els canvis d'elevació perquè, si utilitza ACAD 2000 o 2004 i el seu equip ja li permet de veure correctament el contingut de les finestres des de la modalitat **estructura alámbrica 2D**, només li caldrà anul·lar la línia esmentada. I ara, la pregunta del milió: ¿com podrà saber l'usuari si el seu equip necessita o no **MODOSOMBRA Oculta** per *Assegurar-se una visualització correcta*? Ben senzill: neutralitzant la línia amb el ";" inicial, i provant el programa amb l'opció A i un SUDOKU-PROBLEMA tret de qualsevol lloc, un parell de vegades amb el mètode 1 i un altre parell amb el mètode 2; si no s'observa cap anomalia, l'haurà encertada neutralitzant la línia; si no, l'haurà de reintegrar al codi, treient el ";" inicial, i haurà d'acceptar que les actualitzacions de l'escaquer 9×9 siguin més lentes. Amb aquesta precisió, més que matisar, rectifiquem allò que havíem afirmat a cavall de les pàgines 469 i 470, que va ser escrit abans de la revisió que justifica els apèndixs en negreta de l'últim capítol: "preferim donar per bona la versió que hem ofert sencera en el capítol precedent i limitar-nos ara a indicar quins canvis hauran d'introduir en el codi aquells usuaris que, en executar-la, s'hagin trobat que la resposta del seu equip presenta alguna de les anomalies descrites". Repetim-ho, **sobre la base de la versió completa del codi que clourà aquest document: els qui treballin amb ACAD 2000 o 2004 hauran d'inutilitzar la línia de la funció PREPGRAF-9*9 que acaba amb el comentari "; Només si és ACAD 2000 o 2004, i és imprescindible." i rodar una mica el programa per apreciar si així va bé, reutilitzant-la en cas contrari; els qui treballin amb ACAD 2011, a més d'inutilitzar aquesta línia amb caràcter definitiu, hauran d'habilitar (esborrant el ";" inicial) les línies que acaben amb el comentari "; Només si és ACAD 2011.**", que començareu a veure a partir d'ara.

Deixarem per al final la problemàtica d'ordenació dels objectes que s'encavalquen, a efectes de la visualització desitjada, i començarem per qüestions més trivials.

La primera cosa que va fer l'autor, en assabentar-se de l'existència de diferents estoigs d'eines adaptats a tipus de feina específics, anomenats espais de treball, va ser l'adopció del conegut com "AutoCAD clàsico" (si us costa de trobar el menú desplegable corresponent, podeu invocar la variable de sistema **WSCURRENT** i donar-li aquest valor) i distribuir les barres d'eines tal com acostumava a fer-ho amb ACAD 2004, per no sentir-se estrany amb la interfície. Però això és molt personal.

Aquesta tendència conservadora a intentar mantenir, en cada nova versió d'AutoCAD, l'aparença de la precedent (encara que la nova li oferís més possibilitats) ha dut l'autor a mantenir esquemes mentals que ja no es corresponen amb la realitat, cosa que molts cops es reflecteix en el lèxic utilitzat. El comentari s'escau pels dos elements de l'interfície d'usuari de què tractarem ara i què la versió espanyola d'Autocad anomena *ventana de línea de comando* i *ventana de texto*. En les primeres versions, només per a MS-DOS (va ser fins a la 10, 11 o 12?), en lloc de finestres de mida i localització modificables a gust de l'usuari, eren àrees inamovibles i que probablement tenien el seu origen en la possibilitat d'usar dos monitors (un de gràfic i l'altre de text) i la necessitat d'oferir una alternativa a la majoria d'usuaris que només en tenien un: a sota de l'àrea gràfica teníem un rectangle que ocupava tota l'amplada útil del monitor (la de l'àrea gràfica més la de l'àrea del que s'anomenava menú de pantalla i que, sortosament, va acabant desapareixent) i tenia l'altura de tres línies de text; quan les opcions d'una ordre ocupaven més de 3 línies o convenia consultar algun fragment anterior del diàleg usuari/AutoCAD calia prémer la tecla de funció **<F1>** (abans de traslladar-ho a la **<F2>**, en haver-se reservat **<F1>** per activar **AYUDA**) i el rectangle format per la unió de barra de menús desplegables + àrea gràfica + àrea de missatges i ordres (la de 3 línies de text) cedia el lloc a una àrea de text de 20 línies, donant la sensació que l'àrea de text s'hagués ampliat de 3 a 20 línies que ho tapaven tot. És per això que, al llarg d'aquesta obra, haureu llegit coses com ara "àrea d'ordres", "àrea de text" (a vegades, "àrea inferior de text") o simplement "finestra de text" en lloc de la traducció literal "finestra de línia de comandament", i també "pantalla completa de text" o simplement "pantalla de text" en lloc de la traducció literal "finestra de text", en considerar l'autor que tenir-la activada igual destorbava el dibuix (encara que no ocupés tota la pantalla) i preferir de tenir-la sempre maximitzada. Fet aquest advertiment, convé aclarir que en ACAD 2004 les dues finestres ja eren independents, dimensionables i flotants, i la de línia d'ordres ja era ancorable en qualsevol dels 4 costats, però que en ACAD 2011 hi ha algunes diferències:

- La finestra inferior de text, a banda que l'altura mínima d'una línia de text s'hagi ampliat a dues, la podem tancar (desancorant-la i fent clic a la "X" de la pestanya, o mitjançant l'ordre **OCULTARLINEACOM** o **<Ctrl+9>**). Per aquest motiu, en previsió que en el dibuix en què es treballa s'hagués prescindit d'aquesta finestra, abans de l'ajust sol·licitat pel missatge *Deixeu tres línies de text inferiors [Per seguir, polseu qualsevol tecla]* convé executar l'ordre **LINEACOM**.

- L'altra finestra de text (la no ancorable, que s'activa amb l'ordre **PANTTEXT** i commuta prement **<F2>**) se segueix podent desplaçar (arrossegant per la pestanya superior) i dimensionar (estirant per qualsevol dels 4 costats), o bé maximitzar directament amb el botó *ad hoc* situat a l'extrem dret de la pestanya (entre el de replegament a la barra de tasques de Windows i el de tancament "X"), però, a diferència d'ACAD 2004, una doble commutació des d'una finestra maximitzada amb el botó *ad hoc* no ens retorna al mateix estat, sinó a la situació i dimensions de la finestra flotant prèvia.

El comportament d'aquesta segona finestra en la versió 2011 és força capriciós, ja que, si es vol que la pantalla de text ho cobreixi tot i que després de passar per pantalla gràfica torni a cobrir-ho tot, cal maximitzar-la "manualment", estirant-la per tots ela costats. Però fins i tot així, s'ha de tenir cura de no establir contacte amb els límits superior o inferior (situació en què es percep com una ona que es propaga cap al centre de la pantalla) sinó de quedar-s'hi arran. Suposarem que no hi ha cap altre procediment (si hi ha alguna variable de sistema que reguli el comportament de la finestra de text, l'autor no l'ha sabut trobar), i això ens obligarà a desdoblar el missatge

Amplieu al màxim la finestra de text [Per seguir, polseu qualsevol tecla]

en dos, que amb poques paraules suggereixin com s'ha d'actuar:

Deixeu maximitzada la finestra de text [Per seguir, polseu qualsevol tecla]

per a ACAD 2004, i
Amplieu manualment la finestra de text [Per seguir, polseu qualsevol tecla]
per a ACAD 2011.

Pel que fa a l'anomenada finestra inferior de text, l'execució prèvia de **LINEACOM** solucionarà el problema, però, si efectivament aquesta finestra s'havia suprimit i l'ordre ens la retorna (ACAD 2011), apareixen uns missatges que poden desconcertar l'usuari, abans de la primera expressió a l'espera d'una entrada en temps real (en el nostre cas **(grread)**): ***Cancelar*** (a continuació del missatge *Amplieu manualment la finestra de text [Per seguir, polseu qualsevol tecla]*, proposat en el paràgraf precedent) i, dues linies avall, el *prompt* ***Comando:*** . Com que activar la variable de sistema **NOMUTT** només evitaria ***Comando:*** però no ***Cancelar***, inhibint l'aparició dels missatges programats, cosa amb la qual el remei seria pitjor que la malaltia, no se'ns ha ocorregut res millor que advertir l'usuari sobre aquesta contingència: en comptes de la indicació *... [Per seguir, polseu qualsevol tecla]*, podríem posar *... [Per seguir, polseu qualsevol tecla, encara que surti *Cancelar* i Comando:]*.

Però de seguida veurem que el mateix pot succeir en altres situacions (i no només amb ACAD 2011), així que serà més prudent esperar a saber quants missatges caldria ampliar, per veure si aquesta és la millor solució o serà més entenedor llançar un avís previ amb caràcter general.

Amb ACAD 2011 passa una altra cosa, que ja passava treballant amb ACAD 2004 però que hem preferit tractar ara, just després de comentar l'anomalia que es produeix quan **LINEACOM** té ocasió d'activar una àrea d'ordres prèviament desactivada i la manera més realista d'afrontar-la: la mateixa irrupció sobtada de ***Cancelar*** i de ***Comando:*** es produeix quan la tenim infradimensionada o sobredimensionada (assumim que ja està ancorada a la part inferior) i estirem la vora superior per ajustar-la a tres línies de text; això ja no és privatiu d'ACAD 2011 sinó que es dóna en les dues versions (i suposem que en les intermèdies, si més no).

Pel que fa a la finestra de diàleg de l'ordre **OPCIONES**, cal assenyalar que només desencadenen aquesta resposta (en ACAD 2004 i 2011) determinades opcions entre les quals no es troben les de la pàgina **Visual.**, requadre **Elementos de presentación**, encara que hi haguem d'accedir des d'una altra pègina, però convé estar preparats pel cas que l'usuari aprofiti l'accés a aquesta ordre per activar/desactivar, per exemple, la casella **OPCIONES > Visual. > [Elementos de ventana]: Mostrar barras de desplazam. en la ventana de dibujo**, opció aquesta que sí que porta cua. Molt rebé, però ja que hem comentat que les opcions que ens convida a canviar, si s'escau, el missatge *3) Obriu la pàgina "Visual." i en el requadre "Elementos de presentación" deixeu activada únicament la 1ª casella [Per seguir, sortiu amb "Aceptar"]* són de les innòcues als efectes de què parlàvem, ha arribat el moment d'adonar-nos que hem estat jugant amb foc perquè, tal com fins ara teníem la funció **PREPGRAF-9*9**, els canvis efectuats a **OPCIONES** no arribaríen a temps i podrien causar un bunyol: imagineu-vos, per exemple, el terrabastall que organitzaria SUDÒKULUM en un dibuix amb la casella **OPCIONES > Visual. > [Elementos de presentación]: Crear ventana en nuevas presentaciones** activada; desactivada la variable de sistema **OSMODE**, caldrà desplaçar les ordres/variables **CAPA**, **COLOR**, **TIPOLIN**, **LWDISPLAY**, **ESTILO**, **TILEMODE**, **VENTANAS**, **SCP**, **PLANTA**, **SIMBSCP**, **MODOSOMBRA**, **PRESENTACION**, **PRESENTACION** i **SIMBSCP** (no pas l'última afegida amb ACAD 2011, **LINEACOM**), situant-les després d'**OPCIONES**.

Tot plegat ho veurem 6 pàgines més endavant. De moment, sabent ja que els molestos ***Cancelar*** i ***Comando:*** poden aparèixer en més circumstàncies, optem per deixar les indicacions tal com les teníem abans (però retocant lleugerament altres redactats, per tal que la irrupció dels dos indesitjables mai no interfereixi amb la lectura completa dels textos). En comptes de canvis locals, ampliarem l'anunci dels quatre ajustos amb un advertiment de caràcter general: on fins ara ens limitàvem a posar
Abans que res, cal fer quatre ajustos:
la presentació dirà
*Abans que res cal fer quatre ajustos (si en algun moment surt un *Cancelar* i, dues línies més avall, un Comando: , no en feu cas i seguiu les instruccions):*

A banda del símbol **SCP**, que desactivem a tot arreu, ACAD 2011 disposa d'una eina de navegació anomenada **ViewCube** que, per defecte, apareix en totes les finestres sobre l'Espai Model, a dalt i a la dreta, d'aspecte semitransparent mentre no el trepitgem, i constituïda per un cub i una brúixola. Posarem a **0** la variable de sistema **NAVVCUBEDISPLAY** en cadascuna de les finestres per estalviar-nos la seva incordiant presència: ho fem a l'inici del programa, amb les tres mostres de gris,

i també en les dues finestres on representem l'escaquer 9×9; tot i que en aquestes últimes podria ser que no aparegués, cal preparar-se per al cas més desfavorable. Per què diem això? Doncs perquè *Ayuda de AutoCAD 2011* és prou parca en aquest tema i es limita a dir que *ViewCube es una herramienta de navegación que se muestra al trabajar en un espacio modelo 2D o un estilo visual 3D*, però el comportament de l'eina és bastant capriciós: sí que és veritat que apareix amb un estil visual 3D (i la modalitat **ocUlta** ho és) perquè, si en comptes de **"MODOSOMBRA" "2D"** haguéssim fet **"MODOSOMBRA" "U"**, la bruixoleta sortiria arreu; però amb **estructura alámbrica 2D** només apareix quan l'alçada de la finestra (o de la part que surt en pantalla), just en el moment de crear-la, representa almenys un **1,58**% de l'alçada de l'àrea gràfica de pantalla. Aquesta és la conclusió a què va arribar l'autor després de moltes proves, veient que **ViewCube** només apareixia quan ens presentaven les tres mostres de gris, abans de fer intervenir **NAVVCUBEDISPLAY:** heu de tenir present que en el seu cas les finestres a l'Espai Model, definides per les diagonals que van de -1.25,-0.5 a 0.25,0.5 i de 0.25,-0.5 a 1.25,0.5 (en cordenades d'Espai Paper), abans de fer **ZOOM E** i **ZOOM 0.9X**, quedaven minúscules en un enquadrament d'alçària **VIEWSIZE** = **222.1966** unitats (relació d'alçades, **1**/**222,1966** = **0,45%**); però era clar que, en circumstàncies diferents (haver utilitzat una altra plantilla de dibuix, per exemple, o fer l'àrea gràfica unes altres mides, a causa del monitor o per la disposició de barres d'eines ancorades en pantalla), **ViewCube** podia haver sortit.

Ens estem barallant amb detalls que no deixen de ser secundaris i correm el perill de passar per alt una condició elemental de conseqüències més notòries visualment. Oblidem-nos per un moment de la versió d'AutoCAD i centrem-nos en el format de la pantalla o, més exactament, en la seva relació d'aspecte: totes les il·lustracions que mostren la pantalla corresponen a ACAD 2004 i (això és el que ara interessa) al monitor CRT de proporció **12:9**, i la mida i posició de les dues finestres respon a l'aplicació de **(command "ZOOM" "E" "ZOOM" "0.9X")**, immediatament després d'obrir aquestes finestres en l'Espai Paper (**(command "VENTANAS" "-1.25,-0.5" "-0.25,0.5" "VENTANAS" "0.25,-0.5" "1.25,0.5")**). Doncs bé: per aconseguir que el resultat fos més o menys igual d'armoniós en la pantalla del portàtil, de proporció **16:9** (on ja hem dit que s'havia adoptat l'espai de treball "AutoCAD clàsico" d'ACAD 2011 i ara cal afegir que s'ha respectat la mateixa organització de barres d'eines al voltant de l'àrea de dibuix, hem hagut de fer **(command "ZOOM" "E" "ZOOM" "0.8X");** seguint amb **0.9X**, els rètols al peu de cada finestra s'apropaven massa al límit inferior, i passant a **0.7X**, semblava que desaprofitàvem espai a dreta i esquerra. Però, com que tot això és molt subjectiu, entre els arguments de **command** hem acabat deixant

```
          "ZOOM" "E" "ZOOM" "0.9X"  ; Ajusteu de manera que els retols de peu de
;         "ZOOM" "E" "ZOOM" "0.8X"  ; finestra quedin folgats, en l'àrea grafica.
```

a fi i efecte que cadascú trii el que factor que vulgui o, si li plau, faci **0.85X**.

Aclarit això, allò que primer cridava l'atenció era la conversió a color gris del contorn dels requadres 3×3 en les dues finestres, malgrat haver-se de veure blancs (via PorCapa). Què tenien en comú aquests requadres?: doncs que els de la finestra esquerra pertanyien a la capa NORMAL i els de la dreta a la VARIANT, i que ambdues capes estaven blocades. En realitat, si ens hi fixàvem millor (i dibuixàvem alguna cosa amb el mateix color gris que les línies que formaven la retícula 9×9, però en una capa no blocada), ens adonàvem que les retícules 9×9 també s'havien enfosquit. (En aquest punt, caldria fer una precisió sobre quelcom que fins ara havíem omès: des de l'inici tots els objectes s'han dibuixat en blanc o gris sobre fons negre.) Com que, per poc que remenem ACAD 2011, de seguida descobrirem que entre les noves propietats dels objectes hi ha la transparència, no estranyarà a ningú que semblés que l'atenuació de color observada podia tenir a veure amb això, però no va servir de res posar a **0** la variable de sistema **CETRANSPARENCY**, abans de fer la retícula, ni inhibir la visualització d'aquesta propietat, fent **TRANSPARENCYDISPLAY 0** abans o després. Fins que, perseverant en la cerca (a desgrat i empès per la necessitat, que ja us ha confessat l'autor la poca gràcia que li fa perdre el temps amb totes les foteses -collonades, que deia en Pla- que les empreses de software s'inventen per justificar la vertiginosa successió de versions per poder fer bullir l'olla), va trobar la mare dels ous: per defecte, els objectes dibuixats en capes blocades experimenten una degradació de color mal anomenada *difuminado* (més que un esfumat, és una barreja sostractiva entre el color de l'objecte i el del fons) del 50%, que podem variar o desactivar mitjançant la variable **LAYLOCKFADECTL**. I això és el que hem fet, posant-la a **0** i recuperant l'aspecte de la pantalla gràfica d'ACAD 2004.

En general, si el dispositiu de selecció d'una ordre d'edició captura M objectes, N dels quals pertanyen a capes bloquejades, apareix la confirmació *M encontrados, N estaban en una capa bloqueada* (obviament, l'acció de l'ordre només s'aplica als

M - N). Tanmateix, en ACAD 2011 hi ha almenys una excepció: si una selecció com l'esmentada la fem dins de l'ordre **CAMBPROP**, abans de la confirmació sortiran N advertiments *El objeto está en una capa bloqueada* (un per cada objecte a recer de l'acció de l'ordre); el més greu per a nosaltres és que, si **CAMBPROP** és executada per una expressió AutoLISP, mitjançant la funció **command**, posant a **0** la variable de sistema **CMDECHO** aconseguirem inhibir la confirmació ordinària però no aquests avisos previs. Curiosament això no passa amb l'ordre **CAMBIA**, cosa que permetrà de superar el problema amb la simple substitució de **CAMBPROP** per **CAMBIA Propiedades.** De les diverses aparicions de **CAMBPROP** al llarg del programa, només a tres tindrem aquest efecte indesitjat, consistent en la captura de 60 línies del reticle de 72 situades a la capa "NORMAL" (finestra esquerra), altres 60 a "VARIANT" (dreta) i que, malgrat estar bloquejades ambdues capes i romandre inalterat el seu contingut (a diferència dels caràcters que ocupen les caselles, també capturats), donen lloc a 60 avisos extemporanis: dues se situen en el primer terç de la funció **FES-SUDOKU**

```
(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     **V**V**)
  (if (= 1-2 "1")
      (INI-LLL)
      (progn (NORMTEXT-9*9) (EXPLICACIO)))
  (setq PPM () **V**V** (INI-COORDS)
        NOPLUS (if (= 1-2 "1") **V**V**)
        **V*V** (if (= 1-2 "2") **V**V**)
        GR (if (= 1-2 "1") (strcat "Sobre la VARIANT ESCOLLIDA (dreta), feu clic "
                                   "en les caselles que hagi de veure\nel jugador"
                                   " (per anul•lar els últims clics, polseu la "
                                   "tecla \"<---Backspace\"):") "."))
  (prompt (strcat "\n.\n.\n" GR))
  (if (not (and (= ABC "A") (= 1-2 "1")))
      (command "ESPACIOM" "CVPORT" 2
               "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""))
  (command "CVPORT" 3 "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  ..............................................................
```

i la tercera se situa a la primera meitat de **RASTRE**

```
 (defun RASTRE ()
  (if 1-2
      (progn
        (if (= ABC "A")
            (command "CVPORT" 2 "BORRA" "C" "0,0" "8,8" "" "CVPORT" 3))
        (command "CAPA" "DES" "VAR-1" ""
                 "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-1" ""
                 "CAPA" "ACT" "VAR-1" ""))
      (setq ABC "D"))
  (foreach E PPM (ssadd (ssname (ssget "_C" (car E) (cadr E)) 0) SS))
  (command "CAMBIA" "C" "0,0" "8,8" "E" SS "" "P" "C" (if 1-2 "VAR-0" "NORM-0")
                                                       "L" 0 "")
  ..........................................................................
```

En relació a l'última funció, adoneu-vos que fa poc encara hi teníem una segona **CAMBPROP** conflictiva, just a l'última línia del fragment representat, però havíem decidit de substituir-la per **CAMBIA** a fi de restituir els caràcters seleccionats a l'altura **z** primitiva, cosa que confirma la pertinència de mantenir l'artifici de jugar amb l'**eLev**ació (tot i que amb ACAD 2011 no adoptem el tipus de representació **MODOSOMBRA Oculto**, ara **ocUlto**), per modificar un mínim de línies de codi en passar d'ACAD 2004 (que segueix sent la versió de referència de SUDÒKULUM) a ACAD 2011. Després de la substitució, el mateixos fragments quedaran així:

```
(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     **V**V**)
  (if (= 1-2 "1")
      (INI-LLL)
      (progn (NORMTEXT-9*9) (EXPLICACIO)))
  (setq PPM () **V**V** (INI-COORDS)
        NOPLUS (if (= 1-2 "1") **V**V**)
        **V*V** (if (= 1-2 "2") **V**V**)
```

```
      GR (if (= 1-2 "1") (strcat "Sobre la VARIANT ESCOLLIDA (dreta), feu clic "
                                 "en les caselles que hagi de veure\nel jugador"
                                 " (per anul•lar els últims clics, polseu la "
                                 "tecla \"<---Backspace\"):") ".")) 
  (prompt (strcat "\n.\n.\n" GR))
  (if (not (and (= ABC "A") (= 1-2 "1")))
      (command "ESPACIOM" "CVPORT" 2
               "CAMBIA" "C" "0,0" "8,8" "" "P" "C" "NORM-0" ""))
  (command "CVPORT" 3 "CAMBIA" "C" "0,0" "8,8" "" "P" "C" "VAR-0" "")
  ..............................................................

 (defun RASTRE ()
  (if 1-2
      (progn
        (if (= ABC "A")
            (command "CVPORT" 2 "BORRA" "C" "0,0" "8,8" "" "CVPORT" 3))
        (command "CAPA" "DES" "VAR-1" ""
                 "CAMBIA" "C" "0,0" "8,8" "" "P" "C" "NORM-1" ""
                 "CAPA" "ACT" "VAR-1" ""))
      (setq ABC "D"))
  (foreach E PPM
    (ssadd (ssname (ssget "_C" (car E) (cadr E)) 0) SS))
  (command "CAMBIA" "C" "0,0" "8,8" "E" SS "" "P" "C" (if 1-2 "VAR-0" "NORM-0")
                                                       "L" 0 "")
  ....................................................................
```

Abans de passar a un altre tema, aclarirem per què en la funció **FES-SOLUCIO** no cal canviar **(command "COPIA" "C" "0,0" "8,8" "" "0,0" ""**
"CAMBPROP" "P" "" "C" "VAR-1" "")
L'explicació és senzilla: malgrat que la finestra de selecció de **COPIA** captura 60 línies situades a la capa NORMAL, a més dels 81 caràcters numèrics de la SOLUCIÓ CREADA, només guarda aquests darrers en el conjunt de selecció; de forma que, quan **CAMBPROP** adopta la fórmula **Previo**, sols els 81 caràcters es donaran per al·ludits.

Hem deixat per al final la problemàtica més enfadosa lligada a la transcripció del programa per a ACAD 2011: el control de la visualització (aparició i permanència) dels textos que se succeeixen sobre la franja gris que enllaça les dues finestres. Amb els diversos missatges, compostos per fins a 9 textos simples, hem especificat en quina part del programa s'emeten i des de quina funció es produeix l'accés a la funció **RETOL-2**, vehicle del llançament. Com que 5 dels 7 accesos són a **FES-SUDOKU-2**, hem decidit ordenar-los no pas en funció de la presència en el codi sinó de la seva aparició en el trancurs de l'execució: C) en la preparació de l'inventari de solucions normalitzades compatibles (**PERFIL-V*V** i **COMPAT-PERFILS**); D) si l'usuari desactiva una casella que sortia a **FOTO1**; E) després arribar a una única solució normalitzada compatible; F) en la preparació de l'inventari de totes les solucions compatibles (**ANORMALS**) i G) si l'usuari desactiva una casella que sortia a **FOTO2** (en realitat, torna a sortir el missatge D i a continuació el G). Aquí els teniu:

A) opció A - **MASCA**: *Espereu el missatge "CLIC sobre el nombre..."*

B) mètode 1- **MASCARA** : *Hi ha un procés de cerca en marxa. Per seguir, espereu el símbol del TAO...*

C) mètode 2- **FES-SUDOKU-2**: *Determinar quantes solucions normalitzades són compatibles pot trigar hores. Espereu...*

D) mètode 2- **FES-SUDOKU-2**: *Anul-lant aquesta casella, caldrà recalcular solucions compatibles i tornar a esperar...*

E) mètode 2- **FES-SUDOKU-2**: *Però pot haver-n'hi més, de no normalitzades però igualment compatibles. Així que espereu...*

F) mètode 2- **FES-SUDOKU-2**: *Determinar quantes solucions compatibles hi ha pot trigar una bona estona. Espereu...*

G) mètode 2- **FES-SUDOKU-2**: *...i, si surt "EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA", mai no sabreu si sobren caselles plenes.*

D'entrada, els textos generats per **RETOL-2** no sortien o ho feien amb deficiències: en A, quan apareixien, ho feien incomplets i a batzegades, simultàniament amb els valors candidats acceptats; a B no sortien mai i el fons gris presentava un forat circular en el centre; a C apareixien sempre, tret de la zona circular del centre; a D, E, F i G, quan sortien, també els faltava la zona circular. A partir d'aquí:

- La incorporació de **(command "TEXTOALFRENTE" "T")** a **RETOL-2** es va descartar, per l'aparició de missatges irreductibles a la línia d'ordres (*N objetos puestos delante*, després de cada **RETOL-2**, en els set casos, i de cadascun dels candidats acceptats en el cas A), per no resoldre A ni B, i pels conflictes d'identitat (entre el YIN-YANG i el rectangle gris foradat, a causa de la transferència del referent **uLTimo**) que provocaven efectes perversos en aquest últim cas i en els restants, malgrat la restauració dels textos.
- Incorporant **(redraw)** a **RETOL-2**, l'única diferència era que en A els candidats ja no sortien a batzegades sinó un a un.
- Si, en comptes d'incloure **(redraw)** a **RETOL-2**, es posava **DRAWORDERCTL** a **0** (si no, val **1**), A i B seguien fallant; però a C els textos sortien sencers, i a D, E, F i G ja sortien sempre, i també sencers.
- Incorporant alhora ambdós dispositius (**(redraw)** a **RETOL-2** i **DRAWORDERCTL** a **0**), A ja funcionava correctament i només B es resistia.

Finalment la solució va venir en forma de serendipitat, perquè el que buscàvem era la manera d'evitar la fallada aleatòria de què parlarem a continuació i que s'ha quedat sense resoldre: s'havia pensat en quelcom de més fort que un redibuixat i, vist que **(command "REGEN")** no servia de res (tampoc amb **"REGENT"**), es va provar la sequència **(graphscr) (textscr)**, que tampoc no va servir de res; però transportada a **RETOL-2**, en substitució de **(redraw)**, va ser com posar oli en un llum. Hi havia, però, un inconvenient: la sequència inversa, **(textscr) (graphscr)** (perquè ara érem a la pantalla gràfica i havíem de tornar-hi), es traduïa en un centelleig produït pel pas rapidíssim per la pantalla completa de text sobre fons blanc, que acabava provocant certa fatiga visual. Per minimitzar aquest efecte en va pensar a deixar la funció **RETOL-2** amb el **(redraw)** d'abans, i limitar-nos a incloure la sequència a la funció **MASCARA:**

```
  ..........
  (if 1-2
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")
;       (textscr) (graphscr)                          ; Només si és ACAD 2011.
      )) ...................
```

No vam arribar a fer-ho, perquè abans (sense massa confiança a obtenir resultats) se'ns va ocórrer de substituir-la per **(command "ESPACIOM" "ESPACIOP")** i vam veure que el molest impacte visual ja no es produïa. Qui estigui interessat a escatir la relació causa-efecte entre aquestes dues seqüencies d'ordres i el desblocatge dels textos B que recorri a l'ajut *on line* d'Autodesk o que ho plantegi en algun fòrum, tot i que l'autor l'aconsellaria que abans de res provés les diverses opcions en el seu equip, per allò que havia confessat en abordar aquesta problemàtica d'ACAD 2011: que el tema l'interessava més aviat poc i que tampoc no tenia clar en cada cas si la fallada tenia a veure amb la nova versió d'AutoCAD, amb el nou equip de treball o amb les dues coses alhora.

En relació a aquest tipus de coses, hem deixat per al final la descripció de dues fallades que durant força temps van fer anar de corcoll a l'autor, fins que se'n va adonar que obeïen a la mateixa causa i que aquesta causa transcendia d'AutoCAD. Es podria dir que aquesta descoberta va constituir una serendipitat de retorn, ja que de primer va sorgir l'evidència en el domini **ABC** = **"A"** de **FES-SOLUCIÓ** (SOLUCIÓ CREADA) i després va veure que també era aplicable a **FES-SUDOKU-2** (mètode 2), just a l'inrevés de l'assenyalada a l'últim paràgraf. Deixant ben clar que en aquesta ocasió no es tracta de mostrar la solució i implementar-la en el programa, sinó de saber quines són les condicions que donen lloc al problema, per tal que l'usuari les eviti, començarem per la fallada de **FES-SUDOKU-2.**

Aquest era un territori irredempt sobre el qual l'autor havia efectuat desenes de proves sense sortir-se'n: aconseguir que la funció predefinida **textscr** complís la seva obligació sempre (no sols de tant en tant) quan, havent escollit el mètode 2, el tàndem **PERFIL-V*V** + **COMPAT-PERFILS** responia a l'activació de **POST** no únicament fent inventari de les files compatibles amb cadascuna de les 9 files de la solució escollida, de les tríades de files compatibles entre si, adscrites a cadascun dels

requadres 9×3, i de les combinacions de 3 tríades compatibles entre si, sinó també escrivint puntualment a la finestra de text (de moment reduïda a 3 línies) quants resultats s'obtenien d'aquestes tretze cerques i, sobretot, commutant a pantalla completa de text al final de la segona funció, per poder realitzar una inspecció global i reposada de xifres que només havíem pogut veure desfilar precipitadament. Sobre una pauta dominant de respostes correctes, de manera imprevisible (crèiem) es produïen fallades de tots els colors: que desfilés la sèrie completa de dades, però que en acabar la finestra seguís limitada a tres línies d'ordres i l'usuari s'hagués de molestar a polsar la tecla <F2> per commutar a pantalla de text; que la desfilada s'interrompés amb el nombre tríades normalitzades compatibles amb la primera de la V. E. NORMALITZADA (primer requadre 9×3, a **COMPAT-PERFILS**) però que finalment es desplegués la pantalla de text, mostrant-nos totes les dades; igual encallada, però sense commutació final a pantalla de text; que la interrupció ja es produís després de les 9 compatibilitats amb línies (**PERFIL-V*V**), amb qualsevol dels dos desenllaços precedents (desplegament final o no). Tot i que també s'han produït el que podríem denominar fallades per excés (o, si li poseu imaginació, *poltergeist*), com que la commutació a pantalla completa de text anés acompanyada de fenòmens estranys: activació sobtada de la finestra d'exploració de Windows o d'algun altre programa obert, malgrat que les d'AutoCAD (gràfica i de text) les haguessin tapat; o, més desconcertant encara, reaparició de la finestra de text d'AutoCAD si, per entretenir l'espera, l'usuari havia passat a un altre programa.

Al costat d'aquest desajust n'hi havia un altre, per referir-nos al qual caldrà retrocedir dues pàgines, quan ens referíem als imprevistos que es produïen en els textos que apareixien sobre la franja gris que s'estenia entre les dues finestres i esmentàvem el rètol *Espereu el missatge "CLIC sobre el nombre..."* de l'opció A. Per bé que reconeixíem que aquesta composició de textos generada per **RETOL-2** no sempre sortia i que, quan ho feia, massa sovint era amb deficiències (incompleta i a batzegades, simultàniament amb els valors candidats acceptats), a l'hora de dir que n'hi havia hagut prou a incorporar a **RETOL-2** l'expressió **(command "ESPACIOM" "ESPACIOP")**, no especificàvem si s'havia aconseguit controlar tant el rètol com l'aparició dels valors 1... 9 admesos en les caselles respectives. Perquè encara subsistia una petita anomalia, que vam silenciar: *Espereu el missatge "CLIC sobre el nombre..."* apareixia i desapareixia puntualment, i ho feia sempre; els valors admesos també anaven apareixent puntualment, a mesura que **RE-MEMBER** n'emetia el veredicte de compatibilitat, però no sempre, perquè en activar la primera casella el degoteig d'aquests valors quedava temporalment interromput en moltes ocasions; per concretar, diguem que la fallada consistia en l'aparició de l'**1**, el **2**, el **3**, el **4** i el **5** a intèrvals regulars, seguida de l'estancament del dibuix durant el temps que previsiblement haurien trigat a desfilar-hi el **6**, el **7** el **8** i el **9**, que apareixien de cop després de la pausa. Si no vam voler entretenir-nos precisant aquesta minúcia, va ser perquè els indicis apuntaven més a alguna deficiència de la targeta gràfica i/o del controlador de pantalla que cap a una característica d'ACAD 2011 no tractada adequadament, igual que el descontrol de l'àrea de text que hem descrit en l'últim paràgraf, i teníem previst referir-nos-hi conjuntament.

Mentrestant, però, l'atzar combinat amb un mínim d'intuició a l'hora de relacionar fenòmens aparentment inconnexos ha desvelat la causa d'ambdues anomalies: si, quan haguem creat o carregat el dibuix sobre el qual executarem SUDÒKULUM, activem una finestra aliena a AutoCAD (tant se val que estigui maximitzada, tapant íntegrament la d'AutoCAD, o no) i, de tornada al dibuix, l'execució passa per algun dels punts conflictius assenyalats (en cas d'anar a A per crear un SUDOKU-SOLUCIÓ i després derivar-ne un SUDOKU-PROBLEMA mitjançant el mètode 2, només falla el primer punt), es detectaran les deficiències comentades, que són secundàries i en cap cas posen en risc la correcta execució del programa. Això serà, doncs, gairebé inevitable si optem pel mètode 2, la lentitud del qual en el moment crític **PERFIL-V*V** + **COMPAT-PERFILS** ja ve anunciada (potser exageradament i tot, si ens atenim a les darreres versions) pel rètol *Determinar quantes solucions normalitzades són compatibles pot trigar hores. Espereu...*, perquè estarem tentats d'aprofitar el temps fent alguna altra cosa mentrestrant: si aquesta altra activitat ha de compartir amb SUDÒKULUM la potència de l'ordinador, ja podem comptar que, a banda d'augmentar el temps de processat, potser no caldrà donar un cop d'ull de tant en tant a AutoCAD, perquè quan aquest llarg procés conclogui probablement ens en assabentarem per un sobtat canvi en l'ordenació i visualització de les finestres obertes, i fins i tot podem tenir la sorpresa que la finestra que interrompi el nostre entreteniment sigui la d'AutoCAD. Sistematitzarem mínimament les experiències realitzades, suposant que l'accés a finestres alienes a AutoCAD s'ha realitzat abans d'executar SUDÒKULUM:

- Pel que fa a **PERFIL-V*V** + **COMPAT-PERFILS** < **FES-SUDOKU-2**:
  - Si s'obre la finestra d'exploració de Windows:
    - Per les tres línies de text només veurem desfilar les compatibilitats de les 9 files.
    - Quan s'acabi el procés (amb el nombre de solucions compatibles i l'invitació a polsar qualsevol tecla per seguir), poden passar dues coses:
      - Que la finestra de text segueixi reduïda a tres línies.
      - Que hagués commutat a pantalla completa de text, però quedant parcialment coberta per la finestra de l'explorador de Windows, que caldrà desactivar.
    - Aquesta anomalia no afectarà els nous dibuixos que anem obrint dins de la mateixa sessió de treball amb AutoCAD, tant se val que l'explorador s'hagi activat exclusivament per a una inspecció visual o s'hagin obert carpetes, seleccionat arxius i s'hagin esborrat, retallat, copiat o enganxat.
  - Si s'obre la finestra d'algun altre programa:
    - Per les tres línies de text només veurem desfilar les compatibilitats de les 9 files.
    - Quan s'acabi el procés (amb el nombre de solucions compatibles i l'invitació a polsar qualsevol tecla per seguir), poden passar dues coses:
      - Que la finestra de text segueixi reduïda a tres línies.
      - Que hagués commutat a pantalla completa de text, però quedant visualment coberta per la finestra del programa activat, que haurem de desactivar.
    - Aquesta anomalia no afectarà els nous dibuixos que anem obrint dins de la mateixa sessió de treball amb AutoCAD, tret que hi hagués hagut intercanvi d'informació amb el disc o qualsevol dispositiu de memòria externa (obrir o guardar): en aquest cas l'anomalia també es donarà en dibuixos successius i, si es vol evitar, caldrà tancar AutoCAD i tornar-lo a obrir.

- Pel que fa a **MASCA** < **MASCARA** < **FES-SOLUCIO**:
  - Si s'obre la finestra d'exploració de Windows o d'algun altre programa:
    - En activar la primera casella (per ser exactes i en atenció a modificacions que encara hem d'introduir, la primera on tots els valors de **(ELEMENT LLL)** siguin processats per **RE-MEMBER**), la desfilada dels valors admesos patirà una interrupció, abans d'aparèixer tots.
    - La resta de caselles ja no experimentarà aquesta anomalia, tret que en el transcurs dels emplenaments es tornés a activar alguna finestra.
    - Aquesta anomalia no afectarà els nous dibuixos que anem obrint dins de la mateixa sessió de treball amb AutoCAD, tant se val les operacions que haguem efectuat amb l'explorador de Windows o l'intercanvi d'informació que pugui haver-se produït entre el programa activat i qualsevol dispositiu de memòria externa.

És molt probable que aquestes fallades només es produeixin en configuracions com la de l'autor (targeta gràfica Nvidia Geforce GT330M de 1 GB, amb controlador de pantalla Windows Kernel Mode Driver V. 189.21) i altres equips no en tinguin cap o les tinguin diferents, cas en què l'usuari n'haurà d'indagar l'origen, si són de prou entitat com per justificar l'esforç.

Tot seguit teniu aquelles funcions en què la implementació en ACAD 2011 requereix un tracte diferenciat. Seguirem les normes establertes a la pàgina 500, quant a prevalença de l'adequació a ACAD 2004 i la presència de comentaris al respecte, a a més a més dels quals subratllarem el codi implicat.

```
(defun TRIA-G ()
  (command "DESHACER" "M"
           "VENTANAS" ""
           "ESPACIOM"
;          "NAVVCUBEDISPLAY" 0                          ; Només si és ACAD 2011.
           "ZOOM" "C" "7,4.45" 15.8)
  (repeat 3 ..................) ...)

(defun PREPGRAF-9*9 ()
  (setq G 251               ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        SVT (getvar "SAVETIME")
;       HPD (getvar "HPDRAWORDER")                      ; Només si és ACAD 2011.
;       DRA (getvar "DRAWORDERCTL")                     ; Només si és ACAD 2011.
```

```
;       LAY (getvar "LAYLOCKFADECTL")                    ; Només si és ACAD 2011.
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
;          "LINEACOM"                                    ; Només si és ACAD 2011.
;          "LAYLOCKFADECTL" 0                            ; Només si és ACAD 2011.
;          "DRAWORDERCTL" 0                              ; Només si és ACAD 2011.
;          "HPDRAWORDER" 1                               ; Només si és ACAD 2011.
           "SAVETIME" 0
           "SORTENTS" 127
           "FILLMODE" 1
           "OSMODE" 0 _)
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvingut/Benvinguda al programa SUDÒKULUM!")
  ....................................................................
  (repeat 68 (prompt " "))
  (prompt "Joan Colom\n\n\nAbans que res cal fer quatre ajustos (si en algun ")
  (prompt "moment surt un *Cancelar* i,\ndues línies més avall, un Comando: , ")
  (prompt "no en feu cas i seguiu les instruccions):\n\n1) ")
  (prompt "Deixeu maximitzada")
; (prompt "Amplieu manualment") ; Substitueix la precedent, només si és ACAD 2011.
  (prompt " la finestra de text [Per seguir, polseu qualsevol tecla] ")
  (grread)
  (graphscr)
  (prompt "\n.\n2) Deixeu a baix de tot 3 línies de text  ")
  (prompt "[Per seguir, polseu qualsevol tecla] ")
  (grread)
  (prompt "\n.\n3) Obriu la pàgina \"Visual.\" i en el requadre \"Elementos de ")
  (prompt "presentación\"\n   deixeu activada únicament la 1ª casella ")
  (prompt "[Per seguir, sortiu amb \"Aceptar\"]\n")
  (command "OPCIONES" _
           "CAPA" "E" "PAPER" "I" "~PAPER"
;                 "C" "NOR-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "O" 7       ; ACAD 2000
                  "CR" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "CO" 7    ; ACAD 2004
                  "PAPER,NORM-VAR,NORM-1,VAR-1" "D" "NORM-VAR" ""
           "COLOR" "PORCAPA" "TIPOLIN" "D" "CONTINUOUS" ""
           "LWDISPLAY" 0
           "ESTILO" "STANDARD" "TXT" 0 1 0 "N" "N" "N"
           "TILEMODE" 1 "VENTANAS" "N"
           "SCP" "" "PLANTA" "" "SIMBSCP" "DES" "2D"
           "PRESENTACION" "N" "SUDOKU"
           "PRESENTACION" "D" "SUDOKU"
           "SIMBSCP" "DES")
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: on el color GRIS ")
  (prompt "es diferencii bé\n   del BLANC i el NEGRE, i doni bon contrast entre ")
  (prompt "text i fons.     ")
  (TRIA-G)
  (DIB-3*3 T)
  (command
;          "MODOSOMBRA" "O"   ; Només si és ACAD 2000 o 2004, i és imprescindible.
           "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
           "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
           "CAPA" "B" "NORM-VAR"
;                 "C" G "NORM-0,VAR-0" "D" "PAPER" ""                 ; ACAD 2000
                  "CO" G "NORM-0,VAR-0" "D" "PAPER" ""                ; ACAD 2004
           "PRESENTACION" "D" "SUDOKU"
           "VENTANAS" "-1.25,-0.5" "-0.25,0.5"
           "VENTANAS" "0.25,-0.5" "1.25,0.5"
           "ZOOM" "E" "ZOOM" "0.9X"  ; Ajusteu de manera que els retols de peu de
;          "ZOOM" "E" "ZOOM" "0.85X" ; finestra quedin folgats, en l'àrea grafica.
           "ESPACIOM"
;          "CVPORT" 2                                            ; ACAD 2000
           "CVPORT" 2 "UCSVP" 0                                  ; ACAD 2004
           "VGCAPA" "I" "VAR-1,VAR-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
;          "NAVVCUBEDISPLAY" 0                                   ; Només si és ACAD 2011.
```

```
;           "CVPORT" 3                                                    ; ACAD 2000
            "CVPORT" 3 "UCSVP" 0                                          ; ACAD 2004
            "VGCAPA" "I" "NORM-1,NORM-0" "" ""
            "ZOOM" "-1.515,-1.515" "7.515,7.515"
;           "NAVVCUBEDISPLAY" 0                                 ; Només si és ACAD 2011.
            "SCP" "" "ESPACIOP"
            "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))

(defun RETOL-2 (T1 T2 T3 T4 T5 T6 T7 T8 T9)
  (setq K 0.5)
  (foreach E (list T1 T2 T3 T4 T5 T6 T7 T8 T9)
    (RETOL-1 (list 0 (setq K (- K 0.1))) E)))
; (command "ESPACIOM" "ESPACIOP")                                ; Només si és ACAD 2011.
)

(defun C:SUDOKULUM (/ ANG TG L1A9 LLISTES CONF ABC 1-2 0*0 3*N NIL*NIL G I J K
                      L LL LLL M N N1 N2 N3 N4 P PPM PX PY Q QX QY SS V X Y OK
                      GR POST OSN FIL SRT SVT ECO TN/R-L TT **9*9** **9**9**
 ;                    HPD DRA LAY                                  ; Només si és ACAD 2011.
                    )
  (setq ANG -0.5 TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9))
  ..........................................................
  (command "OSMODE" OSN
           "FILLMODE" FIL
           "SORTENTS" SRT
           "SAVETIME" SVT
 ;         "HPDRAWORDER" HPD                                      ; Només si és ACAD 2011.
 ;         "DRAWORDERCTL" DRA                                     ; Només si és ACAD 2011.
 ;         "LAYLOCKFADECTL" LAY                                   ; Només si és ACAD 2011.
           "DESHACER" "F")
  (setvar "CMDECHO" ECO)
  (princ))
```

Acabarem amb una precisió relativa a la variable de sistema **HPDRAWORDER**, posada a **1** (el valor per defecte és **3**) sense que, en el moment d'escriure aquestes línies, l'autor recordi si l'afectació obeïa a un propòsit concret i encara vigent o, sols és la presència residual d'una de les moltes provatures que havia arribat a fer. Per si de cas ho mantindrem, vist que no sembla interferir en el bon funcionament.

Consells pràctics per evitar una *executio precox*

Com que la intenció de l'autor és que tothom qui no estigui particularment motivat per empassar-se sencer aquest document pugui convertir-se en usuari del programa, sense més requisits que disposar del codi en negreta amb què rematàvem el penúltim capítol (amb les esmenes de l'últim, si s'escau) i, a tot estirar, haver-se mirat el diagrama que hi figura i llegir-se el que comencem ara, s'ha procurat que les referències inevitables a variables i funcions no es limitin al nom sinó que vagin acompanyades de comentaris que en permetin la ubicació aproximada.

Com anunciàvem en el capítol *Condicions mínimes*, es tracta de retardar tot el que sigui possible l'activació de la variable **POST** i la conseqüent posada en marxa de la maquinària pesant, emplenant les caselles previstes però fent-ho de manera que les condicions per a la intervenció de **RE-MEMBER**, **COMPAT-PERFILS** o (ara) **RE+MEMBER** triguin a produir-se i/o que, quan això s'esdevingui, el seu processat duri menys. Deixant de banda un artifici de programació a l'interior de **SUD-MIN** (funció en què inicialment es fa **POST** = **T** i de seguida es desactiva, tret que es donin aquestes condicions mínimes), el pas de **POST** de **nil** a **T** marca totes les entrades en escena de les tres piconadores. Seguint el diagrama que esmentàvem, d'esquerra a dreta i de dalt a baix, **POST** s'activa en les ocasions següents:

Fase presolució - Opció A:

A.1 Quan es crea una situació, prèvia a la possibilitat d'existència d'un sudoku, en què emplenant més caselles amb valors admissibles en primera instància pot quedar compromesa la resolució (a **T+D<12** < **ACTUALITZA** < **FES-SOLUCIO**, amb **1-2** = = **nil**, i es posa en marxa **RE-MEMBER** < **MASCARA** < **FES-SOLUCIO**).

A.2 Quan es donen les condicions mínimes (a **SUD-MIN** < **ACTUALITZA** < **FES-SOLUCIO**, amb **1-2** = **nil**, i es posa en marxa **RE-MEMBER** < **MASCARA** < **FES-SOLUCIO**).

Fase postsolució - Mètode 1:

1.1 Quan es donen les condicions mínimes (a **SUD-MIN** < **ACTUALITZA** < **FES-SUDOKU-1** < **FES-SUDOKU**, i es posa en marxa **RE-MEMBER** < **MASCARA** < **FES-SUDOKU-1** però, amb **1-2** = **T**, **MASCARA** regitra un nombre d'accessos a **RE-MEMBER** en general inferior als casos A.1 i A.2, perquè els interromp quan troba algun valor admissible diferent al de la solució).

Fase postsolució - Mètode 2:

2.1 Quan es donen les condicions mínimes per primer cop (a **SUD-MIN** < **FES-SUDOKU-2** < **FES-SUDOKU**, amb **V** = **FOTO1** = **FOTO2** = **nil**, i es posa en marxa **COMPAT-PERFILS** < **FES-SUDOKU-2** per inventariar les solucions normalitzades compatibles).

2.2 Quan, després d'inventariar en 2.1 les solucions normalitzades compatibles i d'activar més caselles, si s'escau, únicament en queda una (a **FES-SUDOKU-2** < **FES<SUDOKU**, amb **V** = **FOTO2** = **nil**, **FOTO1** = **T** i **KK** = **1**, i es posa en marxa **RE+MEMBER** < **ANORMALS** < **FES-SUDOKU-2** per inventariar el total de solucions compatibles, amb **NORM=1** = **nil**).

2.3 Quan, després de 2.1 i abans de 2.2, es desactiva una casella prèviament declarada pública i que figurava a **FOTO1** (també a **FES-SUDOKU-2** < **FES<SUDOKU**, amb **FOTO2** = **nil**, **V** = **FOTO1** = **T** i **KK** > **1**, i es posa en marxa **COMPAT-PERFILS** < **FES-SUDOKU-2** per reinventariar les solucions normalitzades compatibles).

2.4 Quan, després de 2.2, es desactiva una casella prèviament declarada pública i que figurava a **FOTO2**, tant és si encara hi ha més d'una solució compatible o si, havent-ne quedat una de sola, se segueixen activant caselles per fer-la més fàcil (també a **FES-SUDOKU-2** < **FES-SUDOKU**, amb **FOTO1** = **nil** i **V** = **FOTO1** = **T**, i es posa en marxa **RE+MEMBER** < **ANORMALS** < **FES-SUDOKU-2** per reinventariar el total de solucions compatibles, amb **NORM=1** = **T**).

Les dues últimes ocasions no les considerarem perquè, havent-hi el dispositiu de detecció i avís, l'usari ja decidirà si se la juga o no. Pel que fa a la magnitud de la tragèdia que pot sobrevenir en un i altra cas, *a priori* només podem afirmar que les situacions estan prou equilibrades: en 2.3

De l'últim grup retindrem el primer cas (2.1) i deixarem de banda la resta:

- Així com endarrerir al màxim el moment 2.1 (haver emplenat el máxim de caselles en produir-se aquest succés) és indubtablement positiu perquè el perfil de files elaborat per **PERFIL-V*V** serà més tancat i menor el nombre de files compatibles,

cosa que es traduirà en un nombre més baix de comparacions en **COMPAT-PERFILS**, de cara al moment 2.2 els interessos divergeixen. Podem anar amb peus de plom i per emplenar cada nova casella fer molts temptejos, activant-ne una i desactivant-la després de prendre nota de la reducció de solucions normalitzades compatibles que l'acció comportava, per acabar decidint-nos (activant-la per segon cop i ja amb caràcter definitiu) per la que proporcionava un avenç més gran: actuant així podem estar segurs que més aviat (és a dir, amb menys caselles emplenades) ens en quedarem amb una de sola i, si és certa una de les conjectures del capítol *... però no hi eren totes les que ho són* (que la ràtio entre el el nombre total de solucions compatibles i el de normalitzades no dóna massa salts), arribarem a una certa quantitat de solucions compatibles. Però una cosa és que valorem un sudoku amb quantes menys caselles millor i l'altra estar disposat a pagar-ne el preu, perquè no podem ignorar que l'exploració de l'escaquer 9×9 amb l'algorisme recursiu **RE+MEMBER** serà tant més llarga com més caselles desactivades quedin en el moment d'executar **ANORMALS**.
- Tampoc no considerarem les dues últimes ocasions, perquè havent-hi el dispositiu de detecció i avís,l'usari ja decidirà si se la juga o no. D'altra banda, sobre la magnitud de la tragèdia que pot sobrevenir en un i altra cas cal suposar que, després d'haver activat diverses caselles més des de l'última avaluació global, quan hi tornem per haver-ne desactivat una, el balanç ens serà favorable perquè partirem d'un grau més elevat de concreció: serà lamentable haver d'inventariar de nou, però si el pas enrera havia estat precedit d'uns quants cap endavant, ja no quedaran tantes solucions compatibles per catalogar.

En qualsevol cas, i ja que ens hem referit a la tàctica de temptejar el rendiment de diverses opcions abans d'emplenar cada nova casella, si l'objectiu de l'usuari fos la consecució d'un sudoku poc poblat (o almenys desprovist d'allò que al final del capítol esmentat anomenàvem redundància oculta), recordarem que el mètode 2 (tant a l'etapa "FOTO 1-2" com a la "FOTO 2-2") és l'únic que aporta informació sobre el pes específic de cada casella en el procés de composició del sudoku. I en direcció oposada, si l'usuari volgués afegir al sudoku certa dosi de redundància, tindria més sentit defugir l'activació de caselles irrelevants mentre no assolís la solució compatible única, moment a partir del qual quedaría clar que qualsevol emplenament és superflu.

Així doncs, dels casos en què **SUD-MIN** investiga si es donen els requisits mínims (les condicions necessàries) perquè floreixi un sudoku estricte, ens quedem només amb A.2, 1.1 i 2.1. Les condicions són:
- ...
- ...
- ...
- ...
- ...

Recordem també que l'argument binari **PREVI** controla en quin moment s'activa **POST**, donades les condicions, perquè es posi en marxa la maquinària pesant. Suposant que sigui una determinada casella emplenada la que completa les condicions mínimes,
- Si això ha succeït creant la solució (cas A.1, amb **PREVI** = **T**), el dispositiu **RE-MEMBER** < **MASCARA** es posarà en marxa quan activem la casella següent. És així perquè l'usuari necessita saber quins valors li pot assignar (valors viables en relació a totes les caselles ja activades i no únicament en relació a les de la fila, columna o requadre 3×3), cosa que òbviament depèn de la posició.
- Si això ha succeït seguint el mètode 1 (cas 1.1, amb **PREVI** = **T**), el dispositiu **RE-MEMBER** < **MASCARA** es posarà en marxa quan activem la casella següent. És així perquè, tot i que aquesta última casella (posició) ja té un valor d'assignació predeterminat per la solució adoptada, als efectes de retornar sobre els passos fets i aïllar el sudoku, interessa saber, si més no, si a la casella se li pot assignar algun altre valor (viable en relació a totes les caselles ja activades i no únicament en relació a les de la fila, columna o requadre 3×3), cosa que òbviament depèn de la posició.
- Si això ha succeït seguint el mètode 2 (cas 2.1, amb **PREVI** = **nil**), immediatament es posarà en marxa el dispositiu **PERFIL-V*V** + **COMPAT-PERFILS**. És així perquè l'inventari de solucions normalitzades compatibles només depèn de les caselles activades (posició i valor).
- ...

.....

Però no hem de perdre de vista el cas A.1, perquè pot ser a **T+D<12** on s'activi la variable **POST**, abans que ho faci a **SUD-MIN**.

.....

Fase presolució - Opció A:

A.1 Quan es crea una situació, prèvia a la possibilitat d'existència d'un sudoku, en què emplenant més caselles amb valors admissibles en primera instància pot quedar compromesa la resolució (a **T+D<12** < **ACTUALITZA** < **FES-SOLUCIO**, amb **1-2** = = **nil**, i es posa en marxa **RE-MEMBER** < **MASCARA** < **FES-SOLUCIO**).

A.2 Quan es donen les condicions mínimes (a **SUD-MIN** < **ACTUALITZA** < **FES-SOLUCIO**, amb **1-2** = **nil**, i es posa en marxa **RE-MEMBER** < **MASCARA** < **FES-SOLUCIO**).

.....

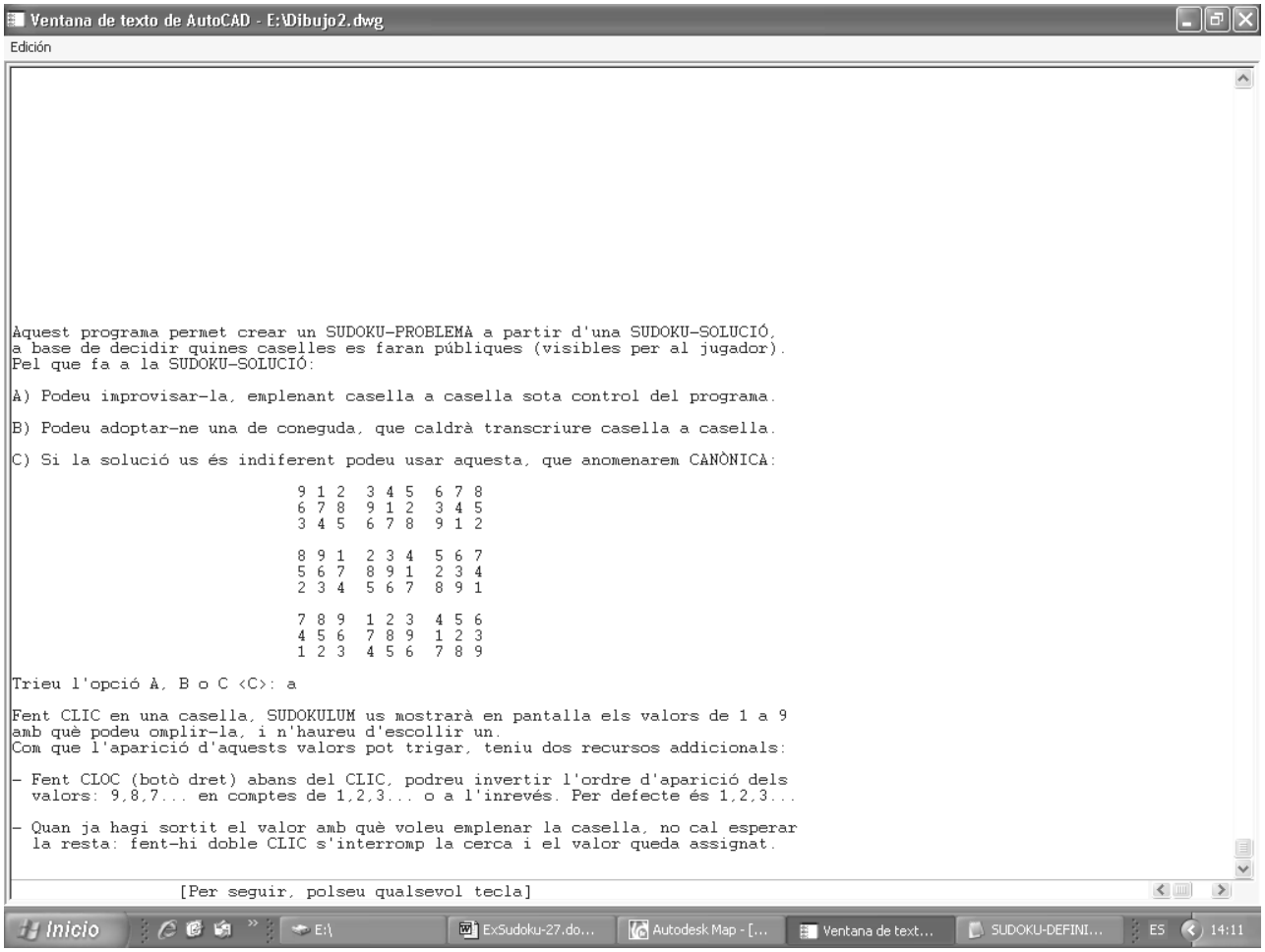
Pel que fa a la fase de presolució, opció A, les primeres versions d'**OMPLE-SUDOKU** (després **FES-SOLUCIO**) encara partien de la creença que el procés d'emplenament de caselles estava caracteritzat per un abans i un després, referits a l'activació de la variable **POST**: mentre **POST** = **nil**, la presentació de candidats era àgil; però el pas a **POST** = **T** comportava l'entrada en escena de **RE-MEMBER**, cosa que pressuposàvem perniciosa per les llargues esperes a què estaria exposat l'usuari. Això, objecte del capítol *Etapa prèvia: crear una solució*, *I (embolica que fa fort)* era veritat, i en aquell context sí que hauria tingut sentit plantejar-nos, com ara hem fet per a les fases postsolució, si hi havia pautes d'actuació en l'emplenament favorables al retardament de l'activació de **POST**. Però ens equivocàvem en creure que hi havia una alternativa fiable a **RE-MEMBER** per identificar els candidats 1... 9 viables, i a *Etapa prèvia: crear una solució*, *II (la difícil simplicitat)* preníem la decisió de partir de **POST** = **T**, adoptant **RE-MEMBER** des del començament i unes mesures tan tímides com ineficaces per posar alguna contenció a la durada que en alguns casos assolia l'execució d'aquesta funció. No ha estat fins a *Etapa prèvia: crear una solució*, *III (roda el món i torna al Born -o a Camprodon-)* que, recuperant per a **RE-MEMBER** alguns dispositius que en els dos capítols precedents havien demostrat no estar a l'altura com a procediments alternatius, hem aconseguit unes reduccions de temps acceptables. En relació a la fase de presolució, per tant, no es tracta tant d'evitar una *executio precox*, *stricto sensu* (i perdó per les dues llatinades) com de dotar l'usuari de recursos addicionals per acabar de llimar les asprors que encara queden en termes d'espera innecessària, ja que, de *precox*, l'*executio* no en pot ser més: a diferència de la fase postsolució en què, tant si seguim el mètode 1 com el 2 disposem de la franquícia atorgada per **SUD-MIN** i cal tenir l'habilitat de postergar l'activació de **POST** fins allà on sigui possible, aquí ja comencem amb **POST** activat i amb la maquinària pesant **RE-MEMBER** a ple rendiment.

De quines asprors parlem? Bàsicament de dues: el *tempo* certament feixuc amb què es van despenjant els valors 1... 9 en el caseller central 3×3, sobretot a l'inici, amb l'escaquer 9×9 encara buit (ja que va *accelerando* a mesura que l'emplenem), i alguna de les pauses escadusseres amb què **MASCARA** encara pot sorprendre i provocar un ensurt, més justificat pel record de les primeres versions (pauses de 10 hores, posteriorment reduïdes a 10 minuts) que per la innocuïtat de les que pot deparar-nos l'actual (en l'últim dels tres capítols esmentats, just abans del codi en què es concretava el contingut argumental, anotàvem una pausa de 10 segons). Doncs bé: atenent particularment a la primera circumstància (perfectament previsible, mentre que l'altra no ho és), presentem dos dispositius complementaris que ens permetran seleccionar el valor pensat per a la casella activada (si és que n'havíem pensat algun, és clar), tan bon punt **RE-MEMBER** el faci aparèixer en el caseller central, sense haver d'esperar l'avís *CLIC sobre el nombre que va a la casella marcada:* que no veurem fins que la funció hagi examinat tots els valors 1... 9 compatibles en primera instància, i fer-ho havent decidit prèviament l'ordre en què convé que ens siguin presentats. Per exemple, si hem decidit assignar un **7** a la casella que anem a emplenar (o, en general, un valor **N ≥ 5**) i **MASCARA** ens presenta els candidats en ordre ascendent (**1** a **9**), potser sortirà a compte invertir l'ordre d'exposició per guanyar temps deturant el procés així que veiem la confirmació d'aquest valor; o, si hem decidit assignar-li un **3** (en general, **N ≤ 5**) i els candidats desfilen en ordre descendent (**9** a **1**), també pot sortir a compte invertir-lo.

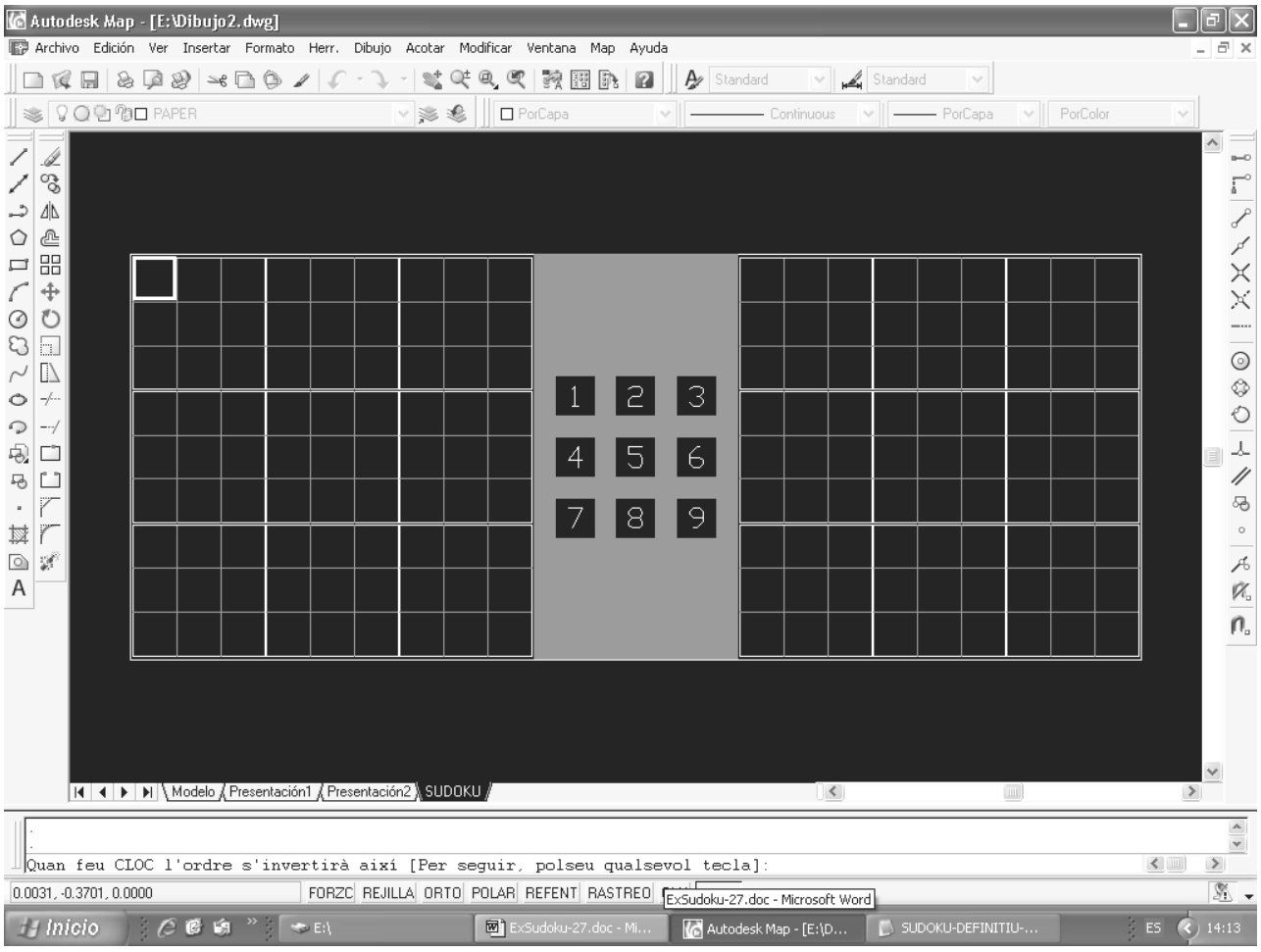
L'opció a invertir l'ordre en què **RE-MEMBER** < **MASCARA** s'ocupa dels candidats i ens els mostra si han superat l'examen la reclamarem fent CLOC (CLIC amb el botó dret) just abans de fer CLIC sobre la casella que volem emplenar (sabrem que és l'ocasió perquè a l'àrea de text hi tindrem el missatge *A l'esquerra, CLIC sobre la casella a omplir:*), i l'efecte serà l'aparició de dues icones damunt i davall del caseller 3×3 (on, per cert, encara hi veurem els candidats de l'última casella emplenada). La icona superior-esquerra és un **1** encerclat per una fletxa orientada cap avall a la dreta (com la diagonal 1-5-9 del caseller), la inferior-dreta és un **9** encerclat per una fletxa orientada cap amunt a l'esquerra (com la diagonal 9-5-1) i caldrà fer CLIC sobre el nombre encerclat que correspongui: si no us heu equivocat i de debò volíeu invertir l'ordre vigent representat per la icona il·luminada, haureu de fer CLIC sobre l'altra icona, que passarà a ser la il·luminada. Quan per fi feu CLIC sobre la casella a emplenar, veureu que els candidats compatibles van fent acte de presència en el caseller 3×3 en l'ordre establert pel canvi, i podreu fer doble CLIC (de seguida justificarem el perquè no podem passar amb un CLIC senzill) sobre el valor previst, tan bon punt aparegui... si és que finalment ho fa, perquè podria haver estat rebutjat per **RE-MEMBER** i no figurar entre els candidats vàlids.

La raó per la qual no podíem abordar la interrupció de l'exploració en curs de **RE-MEMBER** ni la cancel·lació de les iteracions de **MASCARA** amb un CLIC senzill era que només podíem recórrer a un reactiu VisualLISP atès que, per mètodes convencionals, un cop en marxa el motor **MASCARA** l'usuari ja no podia ficar cullerada en el procés fins que l'avaluació del codi no arribés a la funció **grread** instal·lada a **TECLA-P**. I les versions 15 i 16 d'AutoCAD (comercialment, AutoCAD 2000 i 2004), utilitzades en aquest treball, disposaven d'un reactiu **vlr-MOUSE-reactor** associat a la rateta que només era sensible a dos esdeveniments: doble CLIC amb el botó esquerre i CLIC amb el dret (**:VLR-beginDoubleClick** i **:VLR-beginRightClick**). Quant a l'elecció, el doble CLIC semblava menys susceptible de ser confós amb el CLOC d'accés a l'opció a canviar l'ordre de cerca de candidats, permetia romandre amb el dit índex sobre el botò esquerre (després d'haver activat la casella que volíem emplenar) i tenia unes connotacions més resolutives. El CLOC no caldrà associar-lo a aquest reactiu (cosa que encara ens ho complicaria més tot) sinò que el podem integrar a **TECLA-M**, admetent-lo com un valor significatiu més de **(grread)**, que conduirà a l'execució de la nova funció **ORDRE-1-9-1**. És aquí on apareixeran les dues icones adherides al caseller 3×3, haurem de trepitjar-ne alguna per confirmar o invertir l'ordre de cerca vigent (tant aquest, **A-1-9-1**, com el nou, **1-9-1**, poden adoptar els valors **19** -1-9- o **91** -9-1-) i tot seguit tornarem a **TECLA-M** per activar una casella buida. Aquestes icones, la interferència de les quals amb la resta d'elements visuals que es van succeint a la zona central de pantalla (entre les dues finestres obertes en l'Espai Paper) serà menys problemàtica inserint-les com a blocs, podrien crear-se a la mateixa funció **TECLAT** < **FES-SOLUCIO** on bastim el caseller 3×3, però això ens obligaria a preparar-la per evitar conflictes de redefinició en el cas de resposta negativa a la pregunta *Us sembla bé com a solució de treball (S/N)? <S>:* de **CANON->VARIANT** i posterior repetició de l'opció A, raó per la qual hem optat per definir aquests blocs en la funció preparatòria **PREPGRAF-9*9**. També **PREPTEXT-9*9** ha de ser ampliada per incloure-hi una succinta referència a aquests recursos addicionals i una breu *demo* sobre com invertir l'ordre d'aparició dels candidats en el caseller, que a diferència de la definició dels blocs integrarem en **TECLAT** sense problemes.

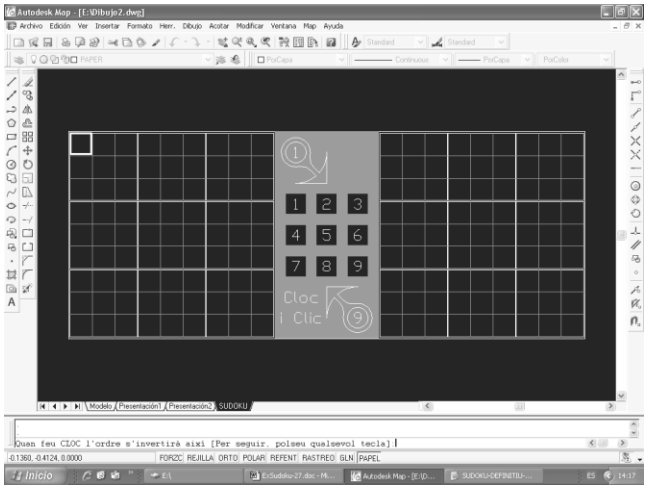
Abans de cloure capítol i volum presentant les funcions noves i les que ha calgut modificar per suportar aquestes possibilitats, farem una puntualització a propòsit de **STOP-MASC**, funció de resposta a l'esdeveniment **:vlr-BeginDoubleClick**, i de ***N***, variable local de **FES-SOLUCIO** que pot semblar supèrflua perquè ja disposem de **N**. Com que, a diferència de les funcions convencionals (en què normalment se sap des d'on són accedides), les associades a un reactiu **vlr-MOUSE-reactor** s'avaluen just quan l'usuari provoca l'esdeveniment (en aquest cas, el doble CLIC), quan hi hagi en joc diverses funcions, jerarquitzades o no però amb variables locals homònimes, i la funció de resposta en manipuli alguna, és difícil de preveure quina s'estarà executant en el moment del succés i correm el risc que la manipulació no afecti la variable que esperàvem sinó una altra: en el cas de **STOP-MASC**, si el valor visible a la casella on fem el doble CLIC s'assignés directament a **N** podria passar que el succés coincidís amb l'avaluació d'**ACTUALITZA-PLUS** (funció que té declarada una **N**) i que, amb independència que aquesta assignació involuntària interferís o no en el resultat de la funció, en tornar a **RE-MEMBER** i finalment a **MASCARA**, **N** recuperés el valor que tenia en aquest àmbit abans que la variable **PASSA** activada per **STOP-MASC** n'interrompés les iteracions. Si voleu estalviar-vos aquest artifici, que comporta reassignar el valor de ***N*** a **N** un cop haguem deixat **MASCARA** enrera, substituïu per qualsevol altre el nom **N** en **ACTUALITZA-PLUS**.



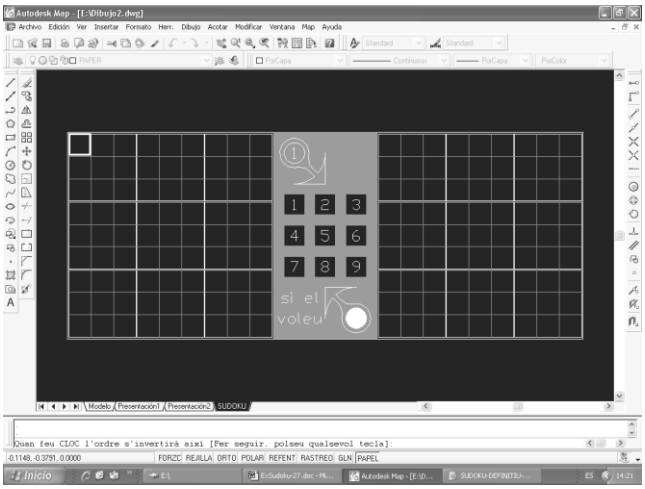
Si escollim l'opció A, se'ns informa que podem fer que la desfilada de valors...



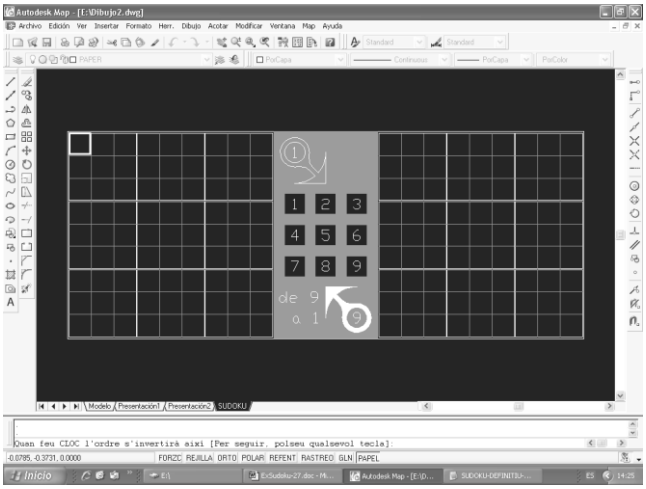
sigui ascendent o descendent, i interrompre-la quan hagi aparegut el que volem.



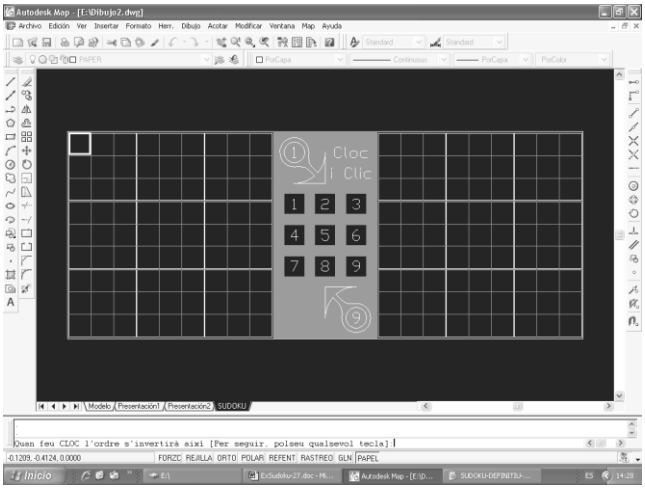
Abans de començar, l'opció A ens mostra què s'ha de fer per invertir l'ordenació dels valors que apareixen en el caseller de la zona central.



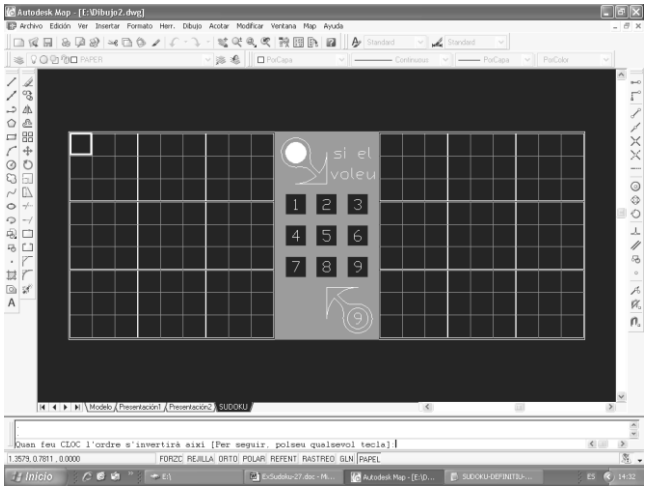
De primer, cal fer cloc (botò dret de la rateta) en aquesta zona. Si després fem clic en el cercle inferior dret...



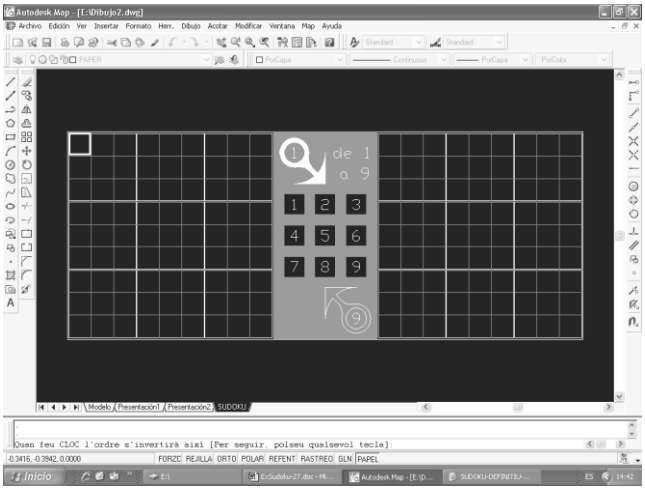
la seqüència numèrica serà descendent (de 9 a 1, pujant pel caseller).



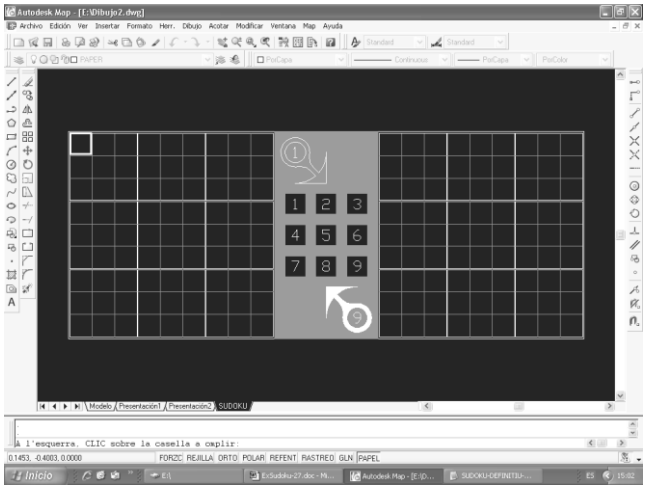
Per contra, si després del cloc fem...



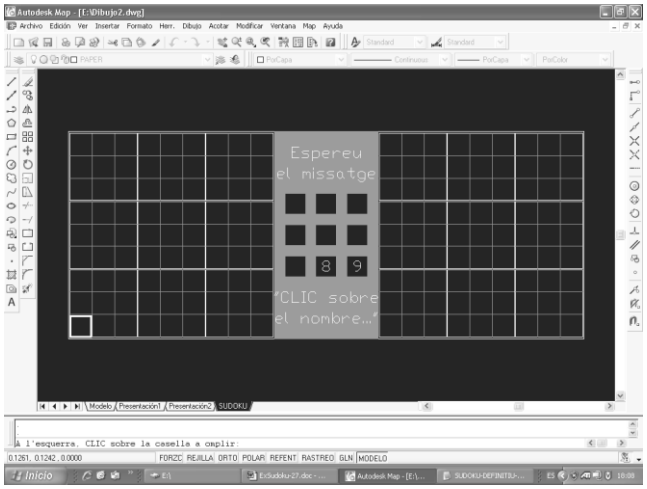
clic en el cercle superior esquerre...



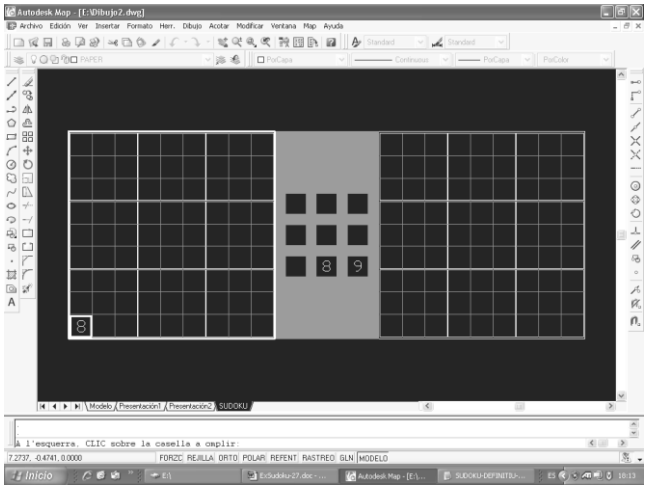
la seqüència numèrica serà ascendent (de 1 a 9, baixant pel caseller).



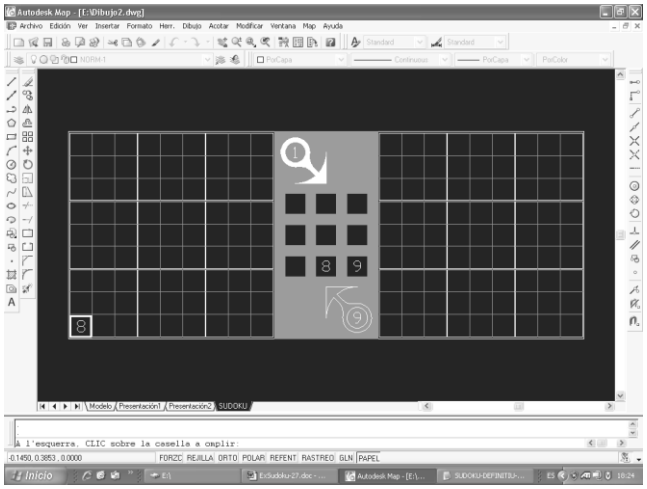
Si sabem que 1,1 és la primera casella que volem activar, i que li assignarem un **8**, farem cloc en la zona central i clic sobre el botò "9".



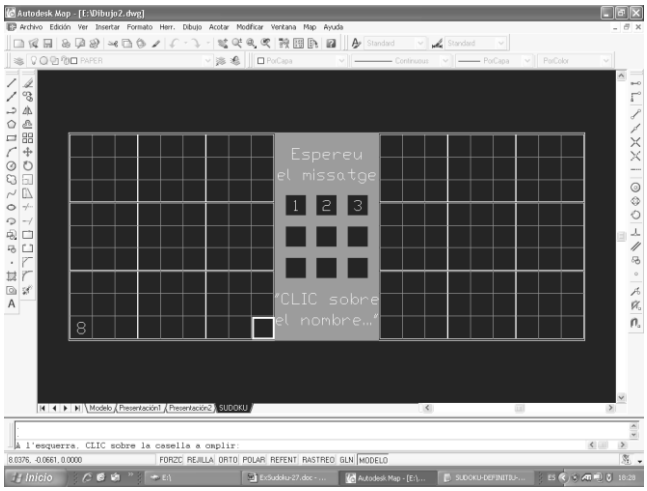
Així, quan fem clic en aquesta casella i apareguin de seguida **8** i **9**, fent un doble clic en l'últim valor ja quedarà assignat, sense haver d'esperar més.



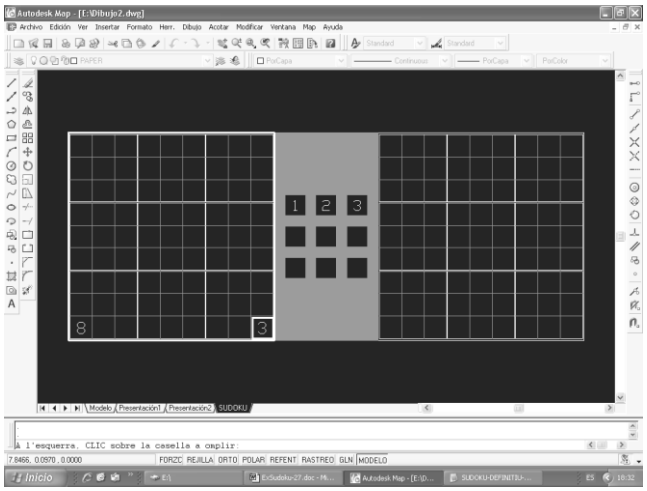
Observeu que per la màscara ja no han desfilat més valors, després del doble clic sobre el **8**.



Si sabem que tot seguit activarem la casella 9,1, i que li volem assignar un **3**, farem cloc en la zona central i clic sobre el botó "1".



Així, quan fem clic en aquesta casella i apareguin de seguida **1**, **2** i **3**, fent doble clic en l'últim valor ja quedarà assignat, sense haver d'esperar més.



Observeu que per la màscara ja no han desfilat més valors, després del doble clic sobre el **3**.

```
(vl-load-com)

(vlr-remove-all)

(vlr-MOUSE-reactor () '((:vlr-BeginDoubleClick . STOP-MASC)))

(defun STOP-MASC (N-REACTIU L-PUNT)
  (if (not PASSA)
      (if (setq GR (list (caar L-PUNT) (cadar L-PUNT)) *N* (TECLA))
          (if (member *N* LL)
              (setq OK T PASSA T)))))

(defun PREPGRAF-9*9 ()
  (setq G 251               ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        SVT (getvar "SAVETIME")
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
           "SAVETIME" 0
           "SORTENTS" 127
           "FILLMODE" 1
           "OSMODE" 0
           "CAPA" "E" "PAPER" "I" "*"
;                 "C" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "O" 7       ; ACAD 2000
                  "CR" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "CO" 7     ; ACAD 2004
                  "PAPER,NORM-VAR,NORM-1,VAR-1" "D" "NORM-VAR" ""
           "COLOR" "PORCAPA"
           "TIPOLIN" "D" "CONTINUOUS" ""
           "LWDISPLAY" 0
           "ESTILO" "STANDARD" "TXT" 0 1 0 "N" "N" "N"
           "TILEMODE" 1
           "VENTANAS" "N"
           "SCP" "" "PLANTA" ""
           "SIMBSCP" "DES"
           "MODOSOMBRA" "2D"
           "PRESENTACION" "N" "SUDOKU"
           "PRESENTACION" "D" "SUDOKU"
           "SIMBSCP" "DES")
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvingut/Benvinguda al programa SUDÒKULUM!")
  (prompt "  *****************")
  (***) (***)
  (prompt "\n\n\nSUDÒKULUM no ha estat concebut per resoldre sudokus (tot i que ")
  (prompt "pot usar-se amb\naquest propòsit, renunciant al goig de jugar-hi com ")
  (prompt "Déu mana) sinó per fer-ne.\n\nFunciona sobre AutoCAD 2004, i supera ")
  (prompt "altres productes similars en que la seva\nlentitud exasperant obliga")
  (prompt "rà l'usuari a exercitar la virtut de la paciència...\n\nConvé aclarir")
  (prompt " que SUDÒKULUM no té res a veure amb CURRÍCULUM ni amb ESPÈCULUM.\n\n")
  (repeat 68 (prompt " "))
  (prompt "Joan Colom\n\n\nAbans que res, cal fer quatre ajustos: ")
  (prompt "\n\n1) Amplieu al màxim la finestra de text [Per seguir, polseu ")
  (prompt "qualsevol tecla]")
  (grread)
  (graphscr)
  (prompt "\n.\n2) Deixeu tres línies de text inferiors [Per seguir, polseu ")
  (prompt "qualsevol tecla]")
  (grread)
  (prompt "\n.\n3) Obriu la pàgina \"Visual.\" i en el requadre \"Elementos de ")
  (prompt "presentación\"\n   deixeu activada únicament la 1ª casella ")
  (prompt "[Per seguir, piqueu en \"Aceptar\"]\n")
  (command "OPCIONES")
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: aquella en què el ")
```

```
  (prompt "color GRIS es\n   diferencii bé del BLANC i el NEGRE, i doni bon ")
  (prompt "contrast entre text i fons.")
  (TRIA-G)
  (DIB-3*3 T)
  (command "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
           "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
           "CAPA" "B" "NORM-VAR"
;                 "C" G "NORM-0,VAR-0" "D" "PAPER" ""                  ; ACAD 2000
                  "CO" G "NORM-0,VAR-0" "D" "PAPER" ""                 ; ACAD 2004
           "PRESENTACION" "D" "SUDOKU"
           "VENTANAS" "-1.25,-0.5" "-0.25,0.5"
           "VENTANAS" "0.25,-0.5" "1.25,0.5"
           "ZOOM" "E" "ZOOM" "0.9X"
           "CIRCULO" "-0.15,0.4" 0.07
           "ARCO" "C" "@" "-0.15,0.25" "0,0.4"
           "LINEA" "-0.15,0.4" "0,0.25" ""
           "DESIGNA" "LT" ""
           "COPIA" "@" "" "M" "@" "@0.01<45" "@-0.01<45" ""
           "BORRA" "P" ""
           "RECORTA" "C" "-0.15,0.4" "0,0.25" "" "DESDE" "0,0.25" "@-0.01<45"
                     "@0.02<45" "@-0.15,0.15" "@-0.02<45"
                     "B" "-0.15,0.4" "0,0.25" "" ""
           "EMPALME" "RA" 0.0793
           "EMPALME" "-0.08,0.4" "DESDE" "-0.075,0.325" "@0.01<45"
           "EMPALME" "-0.15,0.33" "DESDE" "-0.075,0.325" "@-0.01<45"
           "LINEA" "-0.15,0.25" "0,0.25" "0,0.4" ""
           "CIRCULO" "-0.15,0.4" 0.05
           "SOMBREA" "S" "C" "-0.15,0.4" "0,0.25" ""
           "BLOQUE" "1-9-1" "0,0" "LT" ""
           "SOMBREA" "S" "LT" ""
           "BLOQUE" "BOTO" "0,0" "LT" ""
           "MATRIZ" "C" "-0.15,0.25" "0,0.4" "" "P" "0,0" 2 180 "S"
           "TEXTO" "MC" "-0.15,0.4" 0.05 "" "1"
           "TEXTO" "MC" "0.15,-0.4" "" "" "9"
           "BLOQUE" "2x1-9-1" "0,0" "C" "-0.15,-0.4" "0.15,0.4" ""
           "ESPACIOM"
;          "CVPORT" 2                                                   ; ACAD 2000
           "CVPORT" 2 "UCSVP" 0                                         ; ACAD 2004
           "VGCAPA" "I" "VAR-1,VAR-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
;          "CVPORT" 3                                                   ; ACAD 2000
           "CVPORT" 3 "UCSVP" 0                                         ; ACAD 2004
           "VGCAPA" "I" "NORM-1,NORM-0" "" ""
           "ZOOM" "-1.515,-1.515" "7.515,7.515"
           "SCP" "" "ESPACIOP"
           "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))

(defun PREPTEXT-9*9 ()
  (textscr)
  (prompt "\n\nAquest programa permet crear un SUDOKU-PROBLEMA a partir d'una ")
  (prompt "SUDOKU-SOLUCIÓ,\na base de decidir quines caselles es faran públiques")
  (prompt " (visibles per al jugador).\nPel que fa a la SUDOKU-SOLUCIÓ:")
  (prompt "\n\nA) Podeu improvisar-la, emplenant casella a casella sota control ")
  (prompt "del programa.")
  (prompt "\n\nB) Podeu adoptar-ne una de coneguda, que caldrà transcriure ")
  (prompt "casella a casella.")
  (prompt "\n\nC) Si la solució us és indiferent podeu usar aquesta, que ")
  (prompt "anomenarem CANÒNICA:")
  (prompt "\n\n                              9 1 2  3 4 5  6 7 8")
  (prompt   "\n                              6 7 8  9 1 2  3 4 5")
  (prompt   "\n                              3 4 5  6 7 8  9 1 2")
  (prompt "\n\n                              8 9 1  2 3 4  5 6 7")
  (prompt   "\n                              5 6 7  8 9 1  2 3 4")
  (prompt   "\n                              2 3 4  5 6 7  8 9 1")
  (prompt "\n\n                              7 8 9  1 2 3  4 5 6")
  (prompt   "\n                              4 5 6  7 8 9  1 2 3")
  (prompt   "\n                              1 2 3  4 5 6  7 8 9")
  (initget "A B C")
```

```
  (setq ABC (getkword "\n\nTrieu l'opció A, B o C <C>: ") ABC (if ABC ABC "C"))
  (if (= ABC "A")
      (progn
        (prompt "\n\nFent CLIC en una casella, SUDÒKULUM us mostrarà en pantalla")
        (prompt " els valors de 1 a 9\namb què podeu omplir-la, i n'haureu ")
        (prompt "d'escollir un.\nCom que l'aparició d'aquests valors pot trigar,")
        (prompt " teniu dos recursos addicionals:")
        (prompt "\n\n- Fent CLOC (botò dret) abans del CLIC, podreu invertir ")
        (prompt "l'ordre d'aparició dels\n  valors: 9,8,7... en comptes de ")
        (prompt "1,2,3... o a l'inrevés. Per defecte és 1,2,3...")
        (prompt "\n\n- Quan ja hagi sortit el valor amb què voleu emplenar la ")
        (prompt "casella, no cal esperar\n  la resta: fent-hi doble CLIC ")
        (prompt "s'interromp la cerca i el valor queda assignat.\n")
        (SEGUIR ())))
  (graphscr))

(defun MASCARA (L / INI OK PROU C->F Ñ JJ 9**9 0-**9*9** 0-LLL 0-L)
  (if (not 1-2) (setq LL () M 0 SS (ssadd)))
  (if POST
      (progn
        (setq N (N-3*3 (if 1-2 **V**V** **9*9**)) Ñ (TRANSPOSAR N)
              N (1+2+3 N) Ñ (1+2+3 Ñ))
        (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
            (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
        (if (and (>= (car N) (cadr N)) (>= (cadr N) (caddr N)))
            (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL)
            (progn
              (setq JJ (if (and (<= (car N) (cadr N)) (<= (cadr N) (caddr N)))
                           '(2 1 0)
                           (if (>= (car N) (caddr N))
                               (if (> (cadr N) (caddr N)) '(1 0 2) '(0 2 1))
                               (if (> (cadr N) (caddr N)) '(1 2 0) '(2 0 1))))
                    0-LLL (F=3 (if 1-2 **V**V** **9*9**) K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**) 0-LLL (F=3 LLL)
                    P (list X (+ (rem Y 3)
                                 (* 3 (- 3 (length (member (/ Y 3) JJ)))))))))
        (if (< (car Ñ) (caddr Ñ))
            (progn
              (setq K ())
              (foreach EE (reverse 0-**9*9**)
                (setq 0-L ())
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E)))
                                  0-L)))
                (setq K (cons 0-L K)))
              (setq 0-**9*9** K K ())
              (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
              (setq 0-LLL K P (list (- 8 (car P)) (cadr P))))))
  (if (and POST 1-2)
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M"
                 "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")))
  (setq N (if (= A-1-9-1 19) 0 10))
  (while (and (not PASSA) (if (= A-1-9-1 19) (< N 9) (> N 1)))
    (setq N (if (= A-1-9-1 19) (1+ N) (1- N)))
    (if (and POST (not 1-2)) (RETOL-2 "Espereu" "el missatge" "" "" "" "" ""
                                      "\"CLIC sobre" "el nombre...\""))
    (if (and (not (or PROU (= N GR)))
             (member N L)
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
```

```
        (if PASSA
            (if (and POST (not 1-2)) (TREU-RETOL))
            (progn
              (setq LL (cons N LL) M (1+ M))
              (if 1-2
                  (setq PROU T)
                  (progn
                    (if POST (TREU-RETOL))
                    (RETOL-1 (cond ((= N 1) '(-0.15  0.15))
                                   ((= N 2)     '(0  0.15))
                                   ((= N 3)  '(0.15  0.15))
                                   ((= N 4) '(-0.15  0))
                                   ((= N 5)     '(0  0))
                                   ((= N 6)  '(0.15  0))
                                   ((= N 7) '(-0.15 -0.15))
                                   ((= N 8)     '(0 -0.15))
                                   (   T     '(0.15 -0.15)))
                             (itoa (if MASC (nth (1- N) MASC) N)))
                    (command "DESIGNA" (ssadd (entlast) SS) ""))))
        (if (and POST (not 1-2)) (TREU-RETOL))))
  (if POST
      (progn
        (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
        (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
        (setq P (list X Y)))))

(defun PAUSA () (repeat 1500000 (* PI PI)))     ; Ajusteu el valor a conveniència.

(defun TECLAT (/ PRIM I1 I2 ANG T1 T2)
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.2,-0.2" "-0.1,-0.1"
           "COPIA" "LT" "" "-1.0419,0.5919" ""
           "EDITPOL" (setq CURSOR (ssget "_L")) "G" 0.006 ""
           "MATRIZ" "-0.1,-0.1" "" "R" 3 3 0.15 0.15
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "COLOR" G
           "SOMBREA" "S" "C" "-0.22,-0.53" "0.22,0.53" ""
           "COLOR" "PORCAPA")
  (MASCARA L1A9)
  (if (and (= ABC "A") (not RESP) (not 1-2))
      (progn
        (prompt "\n.\n.\nQuan feu CLOC l'ordre s'invertirà així ")
        (prompt "[Per seguir, polseu qualsevol tecla]:")
        (grread)
        (command "INSERT" "2x1-9-1" "0,0" 1 1 0)
        (repeat 4
          (setq PRIM (not PRIM) ANG (if PRIM 180 0)
                I1 (if PRIM "-0.125,-0.3" "0.125,0.4")
                I2 (if PRIM "-0.125,-0.4" "0.125,0.3")
                T1 (if PRIM "de 9" "de 1") T2 (if PRIM "a 1" "a 9"))
          (command "TEXTO" "MC" I1 0.05 0 "  Cloc"
                   "TEXTO" "MC" I2 0.05 0 "i Clic")
          (PAUSA)
          (command "BORRA" "LT" "" "BORRA" "LT" ""
                   "INSERT" "BOTO" "0,0" 1 1 ANG
                   "TEXTO" "MC" I1 0.05 0 "si el"
                   "TEXTO" "MC" I2 0.05 0 "voleu")
          (PAUSA)
          (command "BORRA" "LT" "" "BORRA" "LT" "" "BORRA" "LT" ""
                   "INSERT" "1-9-1" "0,0" 1 1 ANG
                   "TEXTO" "MD" (if PRIM "-0.04,-0.3" "0.21,0.4") 0.05 0 T1
                   "TEXTO" "MD" (if PRIM "-0.04,-0.4" "0.21,0.3") 0.05 0 T2)
          (PAUSA)
          (command "BORRA" "LT" "" "BORRA" "LT" "" "BORRA" "LT" ""))
        (command "BORRA" "LT" ""
                 "ESPACIOM" "CVPORT" 2 "CAPA" "D" "NORM-1" "")
        (setq A-1-9-1 19 1-9-1 19))))
```

```
(defun ORDRE-1-9-1 ()
  (command "ESPACIOP" "INSERT" "2x1-9-1" "0,0" 1 1 0
           "INSERT" "1-9-1" "0,0" 1 1 (if (= A-1-9-1 19) 0 180))
  (while (progn
           (setq 1-9-1
                 (if (or (equal (setq GR (grread)) '(2 13)) (equal GR '(2 32)))
                     A-1-9-1
                     (if (= (car GR) 3)
                         (if (< (distance (setq GR (cadr GR)) '(-0.15 0.4)) 0.07)
                             19
                             (if (< (distance GR '(0.15 -0.4)) 0.07)
                                 91)))))
           (if (and 1-9-1 (/= 1-9-1 A-1-9-1))
               (progn
                 (setq A-1-9-1 1-9-1)
                 (command "GIRA" "LT" "" "0,0" 180)))
           (not 1-9-1)))
  (PAUSA)
  (command "BORRA" "LT" "" "BORRA" "LT" "" "ESPACIOM"))

(defun TECLA-M (/ 1-9-1)
  (while (not (and (= (car (setq GR (grread))) 3)
                   (= (getvar "CVPORT") 2)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (equal (list PX PY) '(4 4) 4.48)
                   (or (< (- PX (setq X (fix PX))) 0.48)
                       (< (- (setq X (fix (1+ PX))) PX) 0.48))
                   (or (< (- PY (setq Y (fix PY))) 0.48)
                       (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                   (ELEMENT LLL)))
         (if (= (car GR) 25) (ORDRE-1-9-1)))
  (setq P (list X Y)))

(defun TECLA ()
  (cond ((equal GR '(-0.15  0.15) 0.05) 1)
        ((equal GR '( 0     0.15) 0.05) 2)
        ((equal GR '( 0.15  0.15) 0.05) 3)
        ((equal GR '(-0.15  0   ) 0.05) 4)
        ((equal GR '( 0     0   ) 0.05) 5)
        ((equal GR '( 0.15  0   ) 0.05) 6)
        ((equal GR '(-0.15 -0.15) 0.05) 7)
        ((equal GR '( 0    -0.15) 0.05) 8)
        ((equal GR '( 0.15 -0.15) 0.05) 9)))

(defun TECLA-P (VARS / FI)
  (while (not (and (or (= (car (setq GR (grread))) 3)
                       (setq FI (and VARS (> (strlen MSG) 51)
                                     (or (equal GR '(2 13)) (equal GR '(2 32))))))
                   (or FI (equal (setq GR (cadr GR)) GR (list (car GR) (cadr GR)))
                               '(0 0) 0.2))
                   (or FI (setq N (TECLA)))
                   (or VARS (= ABC "B") (member N LL)))))

(defun FES-SOLUCIO (/ CURSOR P-CURS PASSA A-1-9-1 *N*)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (setq PPM () **9*9** (INI-COORDS)
              P-CURS '(-1.1919 0.4419) POST T)
        (INI-LLL)
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP"
                        "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
```

```
              (if PASSA (setq PASSA () N *N*) (TECLA-P ()))
              (command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
              (setq PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
                    LLL (ACTUALITZA **9*9** N P LLL T)
                    **9*9** (car LLL) LLL (cadr LLL)))
        (while (= (last (car PPM)) 1) (setq PPM (cdr PPM)))
        (setq POST () SS (ssadd))
        (prompt "\n.\nPodeu seguir, per construir el SUDOKU-PROBLEMA (el joc) ")
        (prompt "a partir d'aquest\nSUDOKU-SOLUCIÓ, o acabar i quedar-vos amb ")
        (prompt "el que heu definit sobre la marxa.")
        (initget "Si No")
        (setq CONF (getkword "\nVoleu seguir (S/N)? <S>: ")
              CONF (if CONF CONF "Si"))
        (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
                 "CAMBPROP" "P" "" "C" "VAR-1" "")
        (if (= CONF "Si") (command "ESPACIOP") (RASTRE)))
      (progn
        (prompt "\n.\n.\nCLIC sobre el nombre que va a la casella marcada: ")
        (setq K 9)
        (repeat 9
          (setq L () J -1 K (1- K))
          (repeat 9
            (setq J (1+ J))
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.110465,0" ""
                     "ESPACIOM"
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N)
                     "ESPACIOP")
            (setq L (append L (list N))))
          (command "DESPLAZA" CURSOR "" "-0.9942,-0.1105" "")
          (setq **9*9** (cons L **9*9**)))))
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))
```

****

****

****

Actualització a l'última versió de FES-SOLUCIO del capítol *6- Etapa prèvia: crear una solució, IV (... o a Camprodon).*

Si no un heu fixat massa en la funció de resposta STOP-MASC, la primera cosa és advertir-vos contra la tentació d'integrar el contingut de la funció TECLA en TECLA-P, com teniem abans: adoneu-vos que en STOP-MASC hi ha un accés a aquesta TECLA segregada.

Pel que fa a la modificació de MASCARA, funció de la qual vam decidir segregar MASCA en l'annex homòleg del capítol 12 (pàgines 466 a 485), això implicarà la substitució de (MASCARA L1A9) per (MASCA L1A9) en la funció TECLAT. Al seu torn, MASCA quedarà així:

```
(defun MASCA (L POST)
  (setq LL () M 0 SS (ssadd)
        N (if (= A-1-9-1 19) 0 10))
  (while (and (not PASSA) (if (= A-1-9-1 19) (< N 9) (> N 1)))
    (setq N (if (= A-1-9-1 19) (1+ N) (1- N)))
    (if (and POST (not 1-2))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                 "el nombre...\""))
    (if (and (member N L)
             (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
             (or (not POST) (setq INI T OK ()) (RE-MEMBER 0-**9*9** 0-LLL)))
        (if PASSA
            (if (and POST (not 1-2)) (TREU-RETOL))
            (progn
              (setq LL (cons N LL) M (1+ M))
              (if (not 1-2)
                  (progn
                    (if POST (TREU-RETOL))
```

```
                    (RETOL-1 (cond ((= N 1) '(-0.15  0.15))
                                   ((= N 2)     '(0  0.15))
                                   ((= N 3)  '(0.15  0.15))
                                   ((= N 4) '(-0.15  0))
                                   ((= N 5)     '(0  0))
                                   ((= N 6)  '(0.15  0))
                                   ((= N 7) '(-0.15 -0.15))
                                   ((= N 8)     '(0 -0.15))
                                   (   T     '(0.15 -0.15)))
                             (itoa (if MASC (nth (1- N) MASC) N)))
                    (command "DESIGNA" (ssadd (entlast) SS) "")))))
        (if (and POST (not 1-2)) (TREU-RETOL)))))

(defun FES-SOLUCIO (/ LLISTES 0*0 NIL*NIL CURSOR P-CURS TP C->F INV-F JJ JJ1 JJ2
                      N/R-L TN/R-L TT PASSA A-1-9-1 *N*)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (INI-LLISTES)
        (INI-LLL ())
        (setq NIL*NIL LLL PPM ()
              0*0 (INI-COORDS) **9*9** 0*0
              P-CURS '(-1.1919 0.4419) POST T)
        (INI-LLL L1A9)
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP"
                        "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (if PASSA (setq PASSA () N *N*) (TECLA-P ()))
               (command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
               (setq PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
                     LLL (ACTUALITZA **9*9** N P LLL T)
                     **9*9** (car LLL) LLL (cadr LLL)))
        (while (= (last (car PPM)) 1) (setq PPM (cdr PPM)))
        ..................................................
        (if (= CONF "Si") (command "ESPACIOP") (RASTRE)))
      (progn .....................................))
  (command "BORRA" CURSOR "C" "-0.22,-0.52" "0.22,0.55" ""))
```

****
****
****

Actualització a l'última versió de FES-SOLUCIO del capítol 7- *Etapa prèvia: crear una solució, V (la Seca, la Meca i la vall d'Andorra)*.

Únicament parlarem de MASCA, no fos cas que l'al·lusió feta en l'annex homòleg del capítol 12 (pàgines 485 a 488) a la simplificació d'un circumloqui innecessari en aquesta funció (també en RE-MEMBER), realitzada al final del capítol 7 (p. 355), a l'hora de substituir

```
  ..............
  (foreach N L1A9
    (if POST (RETOL-2 ...))
    (if (and (member N L)
             (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                 (progn ... RE-MEM)))
        (progn ...)))...
per
  ..............
  (foreach N L
    (if POST (RETOL-2 ...))
    (if (and N (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                   (progn ... RE-MEM)))
        (progn ...)))...
```

fos interpretat com un manteniment del canvi, si ara no insistíssim en què caldrà retornar al circumloqui anterior (no en RE-MEMBER però sí en MASCA). Efectivament, si volem mantenir l'opció que els candidats desfilin del 9 a l'1 (i no sols de l'1 al 9) i l'ús del doble clic per assenyalar un candidat ja validat i interrompre la desfilada, el procediment més simple serà recórrer a while en lloc de foreach, com a motor d'iteracions:

```
(defun MASCA (L POST / RR-9*9 RR-LLL PRIMER CAUSANT CAMI CANVI NOLLL RE-MEM)
  (setq LL () M 0 SS (ssadd))
  (setq N (if (= A-1-9-1 19) 0 10))
  (while (and (not PASSA) (if (= A-1-9-1 19) (< N 9) (> N 1)))
    (setq N (if (= A-1-9-1 19) (1+ N) (1- N)))
    (if (and POST (not 1-2))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                 "el nombre...\""))
    (if (and (member N L)
             (or (not 1-2) (= N GR) (and (< M 2) (or (/= M 1) (>= N GR))))
             (or (not POST) (setq INI T PRIMER T NOLLL () CANVI () OK ())
                 (progn
                   (setq RE-MEM (RE-MEMBER 0-**9*9** 0-LLL))
                   (if CANVI
                       (progn
                         (setq JJ (if (= (/ (cadr NOLLL) 3) 1)
                                      '(1 0 2)
                                      '(2 0 1))
                               9**9 () I -1)
                         (foreach EE (F=3 RR-9*9 T)
                           (setq 0-L () I (1+ I))
                           (foreach E EE
                             (setq 0-L (cons (if (atom E) E (list (car E) I))
                                             0-L)))
                           (setq 9**9 (cons (reverse 0-L) 9**9)))
                         (setq RR-9*9 (reverse 9**9) RR-LLL (F=3 RR-LLL T)
                               PRIMER () CANVI ()
                               RE-MEM (RE-MEMBER RR-9*9 RR-LLL))))
                   RE-MEM)))
        (if PASSA
            (if (and POST (not 1-2)) (TREU-RETOL))
            (progn
              (setq LL (cons N LL) M (1+ M))
              (if (not 1-2)
                  (progn
                    (if POST (TREU-RETOL))
                    (RETOL-1 (cond ((= N 1) '(-0.15  0.15))
                                   ((= N 2)     '(0  0.15))
                                   ((= N 3)  '(0.15  0.15))
                                   ((= N 4) '(-0.15  0))
                                   ((= N 5)     '(0  0))
                                   ((= N 6)  '(0.15  0))
                                   ((= N 7) '(-0.15 -0.15))
                                   ((= N 8)     '(0 -0.15))
                                   (   T     '(0.15 -0.15)))
                             (itoa (if MASC (nth (1- N) MASC) N)))
                    (command "DESIGNA" (ssadd (entlast) SS) "")))))
        (if (and POST (not 1-2)) (TREU-RETOL)))))
```

****

****

****

D'acord amb allò que havíem convingut al final del capítol 7- *Etapa prèvia: crear una solució, V (la Seca, la Meca i la vall d'Andorra)*, us informem que tot el que segueix (inclòs l'*Epileg: això acaba igual que el conte de la sopa de còdols?*) se situa on li pertoca, és a dir que recull les actualitzacions de tots els annexos precedents, reprèn la cronologia habitual i, per tant, recupera la diferenciació gràfica entre tipus normals per al text expositiu i negreta per al codi AutoLISP.

El text d'aquest capítol que precedeix els annexos en negreta, que hem deixat tal com van quedar en iniciar la revisió de la fase de presolució del programa, revela que el retorn al capítol 3 no va ser una decisió presa sense vacil·lacions i en un punt precís del discurs, com d'altre banda és fàcil de comprendre, sinó que es va produir en onades successives. Així heu vist com, de primer, va ser la consciència que les esperes associades al mètode 2 eren inadmissibles i que únicament podrien acceptar-se (amb molta benevolència) si: 1) era possible reduir la freqüència amb què es presentaven, aportant-li a l'usuari una mena de codi de bones pràctiques, i 2) oferíem una alternativa força més solvent, com ara el mètode 1. En relació al segon punt, però, tampoc eren molt brillants els resultats, i no ho diem tant pel mètode 1 en si sinó per l'opció A (SOLUCIÓ CREADA) en què s'havia inspirat aquest mètode i el "mètode 0" (tots dos en el capítol *Un nou camí i també una drecera*), que en l'estadi on s'havia quedat l'opció esmentada (que ja no era el capítol 3 sinó un capítol 5 que encara duia com a títol *Etapa prèvia: crear una solució, III (roda el món i torna al Born -o a Camprodon-)*, perquè encara no havia aparegut la problemàtica de les configuracions de tipus AAA-AAA o AAA-BBB aconsellant separar un capítol *5- Etapa prèvia: crear una solució, III (roda el món i torna al Born)* de *6- Etapa prèvia: crear una solució, IV (...o a Camprodon)*) encara presentava molts flancs desprotegits i que sovint podia sotmetre l'usuari a llargues esperes per conèixer els valors admissibles en cada casella. El doble dispositiu suggerit (decidir si la presentació de candidats admissibles la volíem en ordre ascendent o descendent, i poder interrompre aquesta presentació amb un doble clic) s'oferia com una mesura pal·liativa, que ens hem molestat a adaptar a les versions finals dels capítols 6 i 7, més per una qüestió de complir els compromisos adquirits que per la utilitat que pogués tenir en el context de major eficiència (presentar els candidats a un ritme més viu i minimitzar les plantades) que han assolit aquestes últimes versions (tot i que mai no se sap, perquè també podria ser que el rodatge del programa posés en evidència altres punts febles i tingués sentit recuperar una eina que crèiem obsoleta). Però en relació al primer punt hem d'advertir al lector que la presència abundant de punts suspensius en el text no és cap error, sinó un recurs per indicar que l'hem deixat tal com va quedar abans de retornar al capítol 3 per revisar l'opció A, com a testimoni dels dubtes en què ja es debatia l'autor.

Era prou evident que qualsevol intent de salvar els papers intentant de compensar la tosquetat de l'instrument amb una hàbil manipulació per part de l'usuari estava condemnat al fracàs, perquè unes recomanacions assumibles per no restar llibertat d'actuació ni obligar a estar-ne tant o més pendent que de l'orientació creativa o les intencions exploratòries dificilment podran contribuir a retardar la *executio*, depenent d'uns condicionaments amb tanta lletra menuda com els aplegats a **SUD-MIN**. I encara ho segueix sent perquè, malgrat la petita reducció de l'espera que haguem pogut obtenir en algun cas, per l'acció indirecta de les millores introduïdes a la fase de presolució del programa (amb incidència notable sobre l'acció d'**ANORMALS**) i per haver refinat el disseny de **COMPAT-PERFILS** (amb una incidència mínima sobre aquesta funció), ambdues coses tractades en l'annex al capítol 12 *Actualització a l'última versió de FES-SOLUCIO del capítol 6- Etapa prèvia: crear una solució, IV (... o a Camprodon)*, s'ha de reconèixer que el mètode 2 no és massa homologable. En aquest sentit, l'autor ha de confessar que ha conservat el títol del present capítol perquè li feia gràcia, i per mantenir el tarannà frívol i iconoclasta de la major part de títols que figuren a l'*ÍNDEX*, però cal admetre que ja no respon a uns continguts que no trigareu a veure (tot i que podríem adduir que la intenció última és la mateixa: reconduir els temps d'espera associats als moments crítics del mètode 2 a uns valors acceptables) i que han suposat la superació del dilema.

De fet, hi ha dues estratègies gairebé antagòniques per millorar el rendiment del mètode 2: retardar l'execució de **PERFIL-V*V** i **COMPAT-PERFILS**, retardant algun dels esdeveniments que, més enllà de les 17 caselles emplenades, controlen l'activació de **POST**, mitjançant tècniques que tenen en comú mantenir una distribució irregular dels emplenaments (singularment, mantenint un requadre 9×3 o 3×9 on les ocupacions es produeixin en una única línia); despreocupar-se fins a cert punt de **POST**, però procurant que els emplenaments es distribueixin de la manera més uniforme possible al llarg i ample de l'escaquer 9×9, de forma que, quan **PERFIL-V*V** i **COMPAT-PERFILS** es posin en marxa, la primera funció trobi a la llista **L1** poques files compatibles amb cadascuna de les files actuals, cosa que repercutirà a la segona funció perquè les tríades de files que caldrà considerar en cada requadre 9×3 (per determinar-ne la compatibilitat interna) també seran poques, i perquè el nombre de combinacions de 3 tríades (per determinar quines són compatibles entre si) també serà reduït.

Ambdues estratègies, però, tenen l'inconvenient d'obligar l'usuari a estar pendent de com va distribuint les caselles públiques (amb densitat irregular o uniforme), distraient-lo probablement d'altres criteris que per a ell tenen més importància. I aquí és on a l'autor, que ja donava per bona, resignadament, la versió que heu vist madurar, se li van ocórrer dues possibilitats (independents i compatibles) de millorar les fites assolides:

- Una, referida al mètode 2, que permetia a l'usuari despreocupar-se fins a cert punt de la conveniència de mantenir els requadres 9×3 amb un nombre similar de caselles ocupades, de cara a l'activació de **POST** i conseqüent entrada en escena de la maquinària pesant **PERFIL-V*V** + **COMPAT-PERFILS**. La idea era aplicar a la pseudomatriu 9×9 de treball ****V*V**** una de les manipulacions a què era sotmesa en la fase de presolució ****9*9**** (a l'opció A) o en la de postsolució ****V**V**** (en el mètode alternatiu 1): la transposició. Per bé que aquesta transformació, amb el mètode 2, només tindria lloc quan les diferències d'ocupació entre els requadres 9×3 a la transposada fossin menys acusades que a la ****V*V**** original.
- L'altra, que pot implementar-se com una reforma al mètode 2 o com un mètode més, alternatiu als 1 i 2, consistiria a retardar la posada en marxa de la maquinària pesant, però no per confiar la maniobra a una intel·ligent conducció de l'usuari sinó ajornant l'activació de **POST** directament des de la programació: així, amb més caselles emplenades, **PERFIL-V*V** + **COMPAT-PERFILS** i/o **ANORMALS** (perquè el nou nivell d'ocupació ja permetria sospesar si el segon d'aquests dispositius podria fer-se càrrec de la creació de la llista **SUDOKUS-V*V**, sense passar pel primer) enllestirien la feina abans, en haver menys solucions compatibles (normalitzades o globals). ¿Per què trencar-se el cap amb recomanacions, si l'alternativa era tant evident com apujar el llistó?: en comptes de fer l'inventari de solucions compatibles amb una quantitat reduïda d'ocupacions (17 o poc més), actuació que implica obtenir-ne moltes, fer-ho amb un nombre més gran (25 a 30, per exemple) per obtenir-ne menys i, el més important, per no trigar tant a inventariar-les.

Com que aquesta segona millora ha de comportar avenços més espectaculars en la velocitat de processat però també afectacions més importants (si decidim que no sigui una possibilitat més, sinó que substituiexi l'actual mètode 2), l'abordarem en primer lloc, i deixarem per al final l'eventual transposició de ****V*V**** (que en realitat serà la substitució d'aquesta pseudomatriu pel resultat de normalitzar la transposada de ****V**V****), per escurçar la durada de l'actuació de **COMPAT-PERFILS**.

El particular camí de Damasc de l'autor es va produir massa tard, i la caiguda del ruc (quedaria massa èpic parlar de cavall) va ser aparatosa. Li resulta difícil de precisar el moment, però de segur que va ser mentre intentava reprendre el discurs inicial del present capítol, que havia deixat embastat dos anys enrera (amb certa sensació d'haver-se ficat en un atzucac) per revisar algunes de les incongruències detectades en repassar la intervenció de **SUD-MIN** en el disseny primitiu de l'opció A (l'ús erràtic del senyal **POST**, l'artificiosa dualitat entre les funcions **T+D<12** i **RE-MEMBER**), dins de la fase de presolució del problema, retornant al capítol 3. La llum li va venir en forma de brutal presa de consciència: s'estava crucificant pel seu entossudiment a fer dependre el mètode 2 d'una circumstància lligada a un 0,000.873% de casos, percentatge representat pels 58.259.576.583.290.880 SUDOKUs-PROBLEMA amb SUDOKUs-SOLUCIÓ de 17 caselles (els 47.793 sudokus bàsics de G. Royle multiplicats per les 1.218.998.108.160 variants que n'admet cada un) en relació als 6.670.903.752.021.072.936.960 de possibles, segons B. Felgenhauer i F. Jarvis. I això tindria sentit si no comportés cap mena d'entrebanc, però no sent així no té cap mena de justificació muntar tota l'estratègia sobre un supòsit tan altament improbable. Si hi haguessin quantificacions similars a les dels dos autors citats, que donessin el nombre de sudokus de 18 caselles, 19, 20,... (per cert, ¿algú sap de quantes caselles són els SUDOKUs-PROBLEMA no-redundants més plens?), es podria adoptar, com a nombre d'emplenaments que activés inapel·lablement **POST**, la moda o la mitjana ponderada de la distribució, però no és el cas (a l'autor, que tampoc no s'ha esforçat massa a esbrinar-ho, no li consta), així que deixarem que sigui cada usuari qui decideixi on situa el llistó, mitjançant l'assignació de valor a una variable en funcions de constant, que anomenarem **NUM** i intervindrà a **SUD-MIN**.

En fer l'inventari de solucions compatibles (suposarem que les circumstàncies més favorables ens permetran prescindir de l'artificiosa divisió en dues etapes i anar directament al recompte de totes les solucions; si no fos així ja rectificaríem), quan **POST** s'activi amb un nombre d'emplenaments **n** (**n ≥ NUM** perquè, tot i que més improbable quan més gran sigui **NUM**, encara hi ha la possibilitat que la posada en marxa de la maquinària pesant s'ajorni per incompliment d'alguna de les condicions secundàries de **SUD-MIN**), poden passar dues coses:

- Que n'hi hagi més d'una, i en aquest cas seguirem el camí d'abans: a cada nova casella emplenada, **NUMCOMPAT** ens indicarà quantes de les solucions compatibles inventariades ho segueixen sent; acabarem quan només en quedi una.
- Que només n'hi hagi una, i en aquest cas (a diferència de quan era **n = 17**) pot passar que ja estiguem en presència d'un SUDOKU-PROBLEMA estricte o bé que hi hagi caselles emplenades innecessàriament. Per sortir de dubtes caldrà recular, desactivant caselles en ordre invers a la seva activació, fins arribar a tenir-ne només **17** (encara amb una única solució compatible) o fins que, tenint **m ≥ 17** caselles emplenades, **NUMCOMPAT** informi que hi ha més d'una solució compatible: en el primer cas haurem tingut la insòlita xamba d'ensopegar amb un sudoku mínim (esdeveniment estadísticament impossible, tret que ja sabéssim per on calia anar i ja estigués previst); en el segon, haurem de reactivar la **(m+1)**-èsima casella, perquè el SUDOKU-PROBLEMA correspondrà a **(m+1)** emplenaments.

Una altra qüestió és si, en aquest segon cas, la marxa enrera es podrà fer sempre amb **ANORMALS** o (fins i tot suposant que haguem pogut començar amb aquesta funció quan **m = n**, amb un temps d'espera raonable) pot arribar un moment en què calgui recórrer a **PERFIL-V*V** + **COMPAT-PERFILS**. Aquest temor no seria injustificat perquè, en augmentar el nombre de caselles lliures, augmentaria el nombre i/o longitud de les exploracions (autorecursions) de **RE+MEMBER** i la durada d'**ANORMALS**, no tant a causa d'entrebancs esdevinguts per l'absència d'algunes cauteles que sí que tenia **RE-MEMBER** (però que, al cap i a la fi, estaven previstes per problemes normalment associats a ocupacions encara més baixes) com per la multiplicitat de combinacions de valors a considerar, però hem de pensar que únicament en la primera utilització d'**ANORMALS** (és a dir, en la situació de màxima ocupació) seria necessari detectar totes les solucions compatibles: confirmat que ens espera una dinàmica descendent (només n'hem detectat una i, en conseqüència, el que toca és rebaixar l'ocupació), en les utilitzacions subsegüents n'hi haurà prou a veure si n'hi ha una o més (si n'hi ha més, n'hi haurà prou a detectar l'existència de la segona). De fet, és en el primer cas (trobar-nos més d'una solució i haver de seguir emplenant caselles) quan correm el perill d'haver de suportar una espera molt més llarga amb **ANORMALS** que amb el tàndem **PERFIL-V*V** + **COMPAT-PERFILS**, i la conveniència d'evitar que això passi ens dicta el que hauria de ser la regla d'or del nou procediment: no podent saber *a priori* quantes caselles caldrà activar per arribar a un SUDOKU-PROBLEMA, és millor que **NUM** pequi per excés (deixant-nos en una situació de sobreocupació) que per defecte. Si amb **NUM** o, més ben dit, amb el nombre **n** d'emplenaments amb què s'activa **POST** (perquè hem de recordar la lletra menuda de **SUD-MIN**) l'encertem de ple i arribem directament al SUDOKU-PROBLEMA, perfecte; si la diferència **(n−NUM)** és petita, està bé; si és més gran, tampoc no passa res (sols la petita molèstia de desactivar més caselles); si **(n−NUM)** és negatiu però **(NUM−n)** és petit, haurem d'acceptar una curta espera, però si **(NUM−n)** és més gran, millor deixar-ho córrer.

A la pràctica, l'estratègia a seguir podria ser una aplicació del sistema d'assaig i error consistent a partir d'un valor **NUM** no massa baix (30, per exemple) i veure què passa: si comencem amb una única solució compatible, o amb més d'una però el temps d'espera per saber-ho no és massa llarg (posem que no passa de mig minut), prosseguim; si comencem amb més d'una solució i el temps d'espera és massa llarg (passa de mig minut), cancel·lem l'execució del programa, augmentant el valor **NUM** (de 5, per exemple) i tornem a provar, i així successivament. No cal dir que això només seria acceptable si l'assignació de valor a **NUM** pogués ser interactiva, però així i tot (i donant per suposat que hem modificat el codi, substituint la SOLUCIÓ CANÒNICA per la que ens interessa, perquè reproduir-la cada vegada com una SOLUCIÓ CREADA o COPIADA ja seria inadmissible) resultaria massa feixuc activar a cada nou tempteig les mateixes caselles. És per això que convé assegurar-se de no fer curt, i caldrà corregir per elevació l'aplicació del sistema, tractant de minimitzar les possibilitats d'error. En lloc de començar amb un valor **NUM** no massa baix (havíem recomanat 30), ¿té sentit esperar a haver emplenat les 81 caselles de l'escaquer?: entre poc i massa; abans d'obligar l'usuari a activar-les totes manualment, potser seria preferible trobar-nos l'escaquer 9×9 ple (recordem que el SUDOKU-SOLUCIÓ ja el tenim), però aquesta és una altra història. El que ara interessaria és saber si existeix allò que fa poc anomenàvem els SUDOKUs-PROBLEMA no-redundants més plens; per ser exactes, el que ens interessaria és saber quantes caselles ocupades tenen. A l'autor no l'importa massa reconèixer la seva ignorància sobre aquest particular (ja ho havia fet a la pàgina precedent) perquè, encara que aquest límit superior estigués clarament identificat, adoptar-lo com a valor idoni per a **NUM** tampoc no seria cap garantia de no fer curt: recordem un cop més que el mètode 2 (en el qual estem treballant, tot i no haver decidit si les disquisicions actuals serviran per reformar-lo o per proposar un mètode alternatiu) no el·ludeix la redundància, en no incorporar cap dispositiu tapaforats ni fer inventari de les caselles amb una

única candidatura (valor predeterminat) en el moment de la seva activació, per tal d'esborrar-les del SUDOKU-PROBLEMA. Així que, per no començar amb les 81 caselles plenes, podríem pensar amb els SUDOKUs-PROBLEMA més plens assolits amb el mètode 2 (normalment solucions redundants en el sentit d'haver activat innecessàriament les 9 caselles d'una fila, columna, requadre 3×3 o, en general, d'haver activat alguna casella que ja no admetia cap altre valor que l'assignat) però que s'han deixat en l'estat que tenien en quedar-se amb una única solució compatible, sense afegir més redundància després d'això. En aquest supòsit, ¿a quantes caselles plenes aniríem?

Cabussant-nos en la qüestió i fent mans i mànigues, potser trobaríem un valor per damunt de 64 caselles que funcionés en molts casos, però demostrar la possibilitat teòrica de treure un SUDOKU-PROBLEMA no menys poblat de qualsevol SUDOKU-SOLUCIÓ hauria estat força més prolemàtic. Així que ens plantarem en 64, valor que ja és prou elevat (massa i tot, si pensem que per descobrir un SUDOKU-PROBLEMA correntet de 27 caselles n'haurem d'activar 37 més, per després desactivar-les) però amb una virtut: la seguretat de poder treure un SUDOKU-PROBLEMA amb almenys 64 caselles de qualsevol SUDOKU-SOLUCIÓ. Només cal començar emplenant dos requadres 9×3 sencers i una fila sencera del tercer requadre (o bé dos requadres 3×9 sencers i una columna sencera del tercer requadre): fins aquí encara no haurà aparegut el desitjat avís *EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA*, però si activeu una casella més (la 64), o dues (65) o tres (66) que estiguin alineades i pertanyin al mateix requadre 3×3, l'avis us anunciarà que ja teniu SUDOKU-PROBLEMA. Hem fet nombroses proves amb diferents SUDOKUs-SOLUCIÓ i hem constatat dues coses: la que us acabem de descriure (que, si no n'hi ha prou amb les 64 caselles, només cal afegir-ne una o dues més) i que, si hem hagut de recórrer a 65 o 66, amb alguna altra combinació de dos requadres 9×3 i una fila (o de dos requadres 3×9 i una columna) ja farem bingo a la casella 64. Entenem-nos: aquestes dues observacions no passen de ser una conjectura que potser sigui fàcil de desmuntar, però el que va a missa és la impossibilitat d'arribar a un SUDOKU-PROBLEMA, emplenant caselles com hem explicat, abans d'activar la 64, i no només perquè la funció **SUD-MIN** estigui construïda de manera que **POST** no es pot activar mentre dues files o columnes d'un requadre 9×3 o 3×9 estiguin buides, sinó perquè precisament aquesta condició es va incorporar a la funció per tenir present un enunciat de validesa universal que trobareu en el capítol *Condicions mínimes*: EN UN SUDOKU AMB SOLUCIÓ ÚNICA, ELS REQUADRES 9×3 i 3×9 HAN DE TENIR ALMENYS DUES FILES (SI EL REQUADRE ÉS DE 9×3) O COLUMNES (SI ÉS DE 3×9) OCUPADES, ÉS A DIR, TOTALMENT O PARCIAL PLENES.

9 1 2	**3 4 5**	**6 7 8**
6 7 8	**9 1 2**	**3 4 5**
3 4 5	**6 7 8**	**9 1 2**
8 9 1	**2 3 4**	**5 6 7**
5 6 7	**8 9 1**	**2 3 4**
2 3 4	**5 6 7**	**8 9 1**
7 8 9	1 2 3	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	**4 5 6**	**7 8 9**

9 1 2	**3 4 5**	**6 7 8**
6 7 8	**9 1 2**	**3 4 5**
3 4 5	**6 7 8**	**9 1 2**
8 9 1	**2 3 4**	**5 6 7**
5 6 7	**8 9 1**	**2 3 4**
2 3 4	**5 6 7**	**8 9 1**
7 8 9	**1 2 3**	4 5 6
4 5 6	7 8 9	1 2 3
1 2 3	**4 5 6**	**7 8 9**

9 1 2	**3 4 5**	**6 7 8**
6 7 8	**9 1 2**	**3 4 5**
3 4 5	**6 7 8**	**9 1 2**
8 9 1	**2 3 4**	**5 6 7**
5 6 7	**8 9 1**	**2 3 4**
2 3 4	**5 6 7**	**8 9 1**
7 8 **9**	**1 2 3**	**4** 5 **6**
4 5 **6**	**7** 8 **9**	**1** 2 **3**
1 2 3	**4 5 6**	**7 8 9**

Il·lustrarem el que s'ha dit de certs SUDOKUs-PROBLEMA sobre la SOLUCIÓ CANÒNICA. A l'esquerra la representem amb la fila inferior i amb els requadres 9×3 central i superior totalment ocupats. Si, havent-ho deixat així amb el mètode 2, emplenéssim la casella 4,3 (**1**), **POST** s'activaria i, en una rapidíssima execució de **PERFIL-V*V** i **COMPAT-PERFILS**, se'ns informaria de l'existència de dues solucions normalitzades compatibles. L'activació de 5,3 (**2**) no alteraria la situació però, en activar 6,3 (**3**), passaríem a tenir una única solució normalitzada que **ANORMALS** ens confirmaria com a única solució compatible. El resultat, que representem al mig, és un SUDOKU-PROBLEMA de 66 caselles, però veureu quant fàcilment n'obtenim un de 64: només cal transposar aquesta solució (aplicant la transformació TR) i procedir a l'activació de caselles seguint la mateixa plantilla (deixant la fila inferior i els requadres 9×3 central i superior totalment ocupats); quan activeu la casella 4,3 (on ara hi haurà un **4**), s'us informarà que només hi ha una solució normalitzada, que també és l'unica compatible. No ens hem molestat a representar-ho perquè l'espai a la dreta el reservàvem per a un nou SUDOKU-PROBLEMA, que segurament és el paradigma d'allò que anomenàvem "SUDOKU-PROBLEMA més poblat assolible amb el mètode 2, sense afegir redundància després de convertir-se en única solució compatible": hem fet com a la representació del mig, però amb 10 emplenaments més (en negreta) i deixant 5,3 (**2**) per al final; així completem 76 caselles en arribar a l'única solució compatible.

Ara sí que hem begut oli! Volíem una informació més completa de tot el ventall de possibilitats i ja la tenim: de 17 a 76 caselles plenes, ¿en quina posició deixem el llistó, per tal que l'estratègia alternativa permeti aproximar-nos en un temps raonable a totes les possibilitats, activant o (més aviat) desactivant caselles? Havíem quedat en donar-li a l'usuari l'opció de fixar el valor **NUM**, però l'hauríem d'orientar en la tria, assabentant-lo del ventall disponible i oferint-li un valor per defecte com a elecció més neutra i equilibrada. I, com que ja ens hem prodigat prou explicitant les consideracions que ens ha dut fins aquí, no ens entretindrem més comentant el perquè de les diferents opcions VALOR MÍNIM - VALOR PER DEFECTE - VALOR MÀXIM contemplades. Ens limitarem a consignar que, després de **17 - 47 - 76**, hem anat derivant per **17 - 41 - 66**, **17 - 40 - 64** i **17 - 36 - 64**, fins aturar-nos en **18 - 36 - 54**, i a justificar aquesta última i definitiva opció: a banda de ser valors fàcils de recordar (senzill, doble i triple), 18 ja ens situa prou a prop dels improbabilíssims sudokus mínims i, si mantenim el mètode 2 (provisionalment el mantindrem, anomenant mètode 3 a la variant que estem a punt de desenvolupar), ja hi deixarem el llistó a 17 emplenaments, per tal d'utilitzar-lo per fer proves amb els sudokus de la col·lecció de Gordon Royle; pel que fa al límit superior 54, estem d'acord que ens situa molt lluny dels sudokus de 64, 65 i 66, i encara més dels "màxims" (en el sentit restrictiu que ja hem aclarit) de 76 caselles plenes, però tampoc no necessitem aproximar-nos-hi més, perquè en tots aquesta casos allò que ens permetia arribar a un grau tan elevat d'emplenament no era pas la condició principal (el llistó **NUM** a 64, 66 o 76) sinó l'exigència secundària que no quedés cap requadre 9×3 amb dues files buides ni cap altre 3×9 amb dues columnes buides; quant al valor per defecte 36, el millor argument a favor seu és que en cap de les proves fetes amb sudokus que es publiquen a la premsa no ha resultat inconvenient.

De tota manera, i en relació a l'eventualitat que l'usuari estigués interessat en la recerca de sudokus poblats, del tipus dels representats a la pàgina precedent o amb no tantes caselles ocupades (en no haver recorregut a l'artifici de mantenir buides dues files o columnes d'un mateix requadre, fins ja ben avançat el procés) però igualment amb més de plenes que de buides, no aniria malament incorporar un quart mètode (si hem dit que de moment no tocaríem el 2 i que anomenaríem mètode 3 al que hem perfilat, aquest seria el 4) que presentés l'escaquer 9×9 completament ple i convidés l'usuari a desactivar caselles fins arribar a un SUDOKU-PROBLEMA, com ja havíem apuntat dues pàgines enrera. La mecànica seria la mateixa, però ara podrem estar segurs de no fer curt: únicament quan **NUMCOMPAT** informi que hi ha més d'una solució compatible haurem de reactivar l'última casella desactivada, per tal de retornar al SUDOKU-PROBLEMA. Dit això, veiem la implementació dels dos mètodes.

La primera cosa a tenir en compte és que de moment farem un muntatge exclusivament pensat perquè els anomenats mètodes 3 i 4 funcionin correctament, prescindint dels aspectes formals de conducció de l'usuari a través de les diverses opcions, perquè encara no tenim del tot clar si el més encertat serà substituir el mètode 2 pel 3, pel 3 i el 4 o deixar-los tots (2, 3 i 4, com a variants d'un mètode alternatiu al mètode 1). Així doncs, no tocarem les funcions **NORMTEXT-9*9** ni **EXPLICACIO**, que amb els mètodes 3 i 4 seguiran donant una informació que només procedeix referir al 2, i **TRIA-METODE** únicament serà modificada el allò que sigui imprescindible per posar a prova les innovacions. Gràficament també passarem amb el menor nombre de canvis possible, fins al punt de seguir visualitzant a la finestra de l'esquerra la V. E. NORMALITZADA quan l'activació o desactivació de caselles es doni en el marc dels mètodes 3 o 4, tot i haver decidit finalment prescindir de les funcions **PERFIL-V*V** i **COMPAT-PERFILS** (que eren les úniques que treballaven amb solucions normalitzades compatibles) en el primer (òbviament, en el segon ni ens ho hem plantejat), vistos els resultats satisfactoris proporcionats per **ANORMALS**. No insistim en el caràcter provisional d'aquestes incongruències, que hauran d'esmenar-se així que el nucli dur dels mètodes 3 i 4 estigui resolt. Una altra cosa és si seguirem mantenint el protagonisme de ****9*9**** i ****V*V**** (la V. E. NORMALITZADA i la seva part pública) sobre ****9**9**** i ****V**V**** (la VERSIÓ ESCOLLIDA, sense normalitzar, i la seva part pública), en el sentit de seguir aplicant tots els processos sobre les primeres i traslladar els resultats sobre les segones a efectes de visualització, quan seria més lògic intercanviar els papers, atès que ****9*9**** i ****V*V**** seran utilitzades i visualitzades només en un dels quatre mètodes d'obtenció del SUDOKU-PROBLEMA que a partir d'ara compondran el repertori. Mentre aquesta incongruència no es manifesti de portes enfora i de portes endins no suposi una complicació excessiva del codi, ens ho podríem permetre. Per ara, també en això seguirem la llei del mínim esforç. Aclarides les qüestions prèvies, descriurem les afectacions i us oferirem el codi.

Després d'haver afegit **NUM** a les variables en funció de constants **ANG**, **TG** i **L1A9**, d'assignar-li el valor per defecte **36** (a l'inici de **C:SUDOKULUM**) i de donar-li a l'usuari l'oportunitat de personalitzar-lo (al final de **TRIA-METODE**) si escull el mètode 3, el pas següent és incorporar-lo a **SUD-MIN** per adaptar aquesta funció al nou mètode: en comptes d'activar-se **POST** quan, a més de les condicions secundàries d'emplenament (capítol *Condicions mínimes*), el nombre **I** de caselles ocupades sigui **I ≥ 17**, ho farà quan **I ≥ NUM**, amb l'ajut de la variable local **NOPOST**; a més, des del moment en què **I = 17** obrirem una llista **PPM**, similar a la usada en el mètode 1 (capítol *Un nou camí i també una drecera*), per enregistrar el camí recorregut fins a l'activació de **POST**, a fi de poder-lo recórrer en sentit contrari si hi ha hagut sort i aquesta activació ens deixa amb una única solució compatible. En aquest cas la missió d'**ANORMALS** es limitarà a explorar solucions compatibles fins a trobar-ne 2, a diferència del cas que haguéssim fet curt i calgués seguir activant caselles, en què serà interessant saber quantes n'hi ha, de solucions compatibles, per tal que l'usuari n'estigui informat i pugui actuar en conseqüència (anul·lant l'últim emplenament, si convé): per això hem dotat aquesta funció amb l'argument binari **12<34**, que permetrà decidir entre les dues modalitats d'exploració; amb el mètode 3 caldrà distingir entre dinàmica ascendent o descendent, per assignar-li **T** o **nil**; amb el mètode 4 sempre serà **nil**. Prèviament a tot això, la inicialització de les pseudomatrius 9×9 a **FES-SUDOKU** dependrà del mètode (només ens referirem a ****9**9**** i a ****V**V****, perquè ja hem aclarit que l'existència i visualització de ****9*9**** i de ****V*V****, que en els mètodes 3 i 4 no té cap sentit, era un pegat provisional): de moment, limitem-nos a dir que el mètode 3 comença amb les caselles buides, com els mètodes 1 i 2 (amb els elements de ****V**V**** representats per les coordenades i amb els del SUDOKU-PROBLEMA ****9**9**** dibuixats en gris), mentre que en el métode 4 totes estan plenes (****V**V**** comença amb els elements igualats als de ****9**9**** i dibuixats en negre, tapant els de ****9**9**** dibuixats en gris); una vegada haguem mostrat el conjunt de tot el codi afectat, en tindreu una descripció més acurada.

Hem volgut aprofitar aquesta revisió de **FES-SUDOKU** i la conveniència que l'usuari dels mètodes 3 i 4 se senti acompanyat quan apareix el símbol del Tao, més enllà d'unes explicacions prèvies que encara estan per fer, per proporcionar el mateix acompanyament en el cas del mètode 2: quan l'etapa de postsolució es limitava a aquest mètode (abans del capítol *Un nou camí i també una drecera*), potser refiats de l'eficàcia de **NORMTEXT-9*9** i **EXPLICACIO** (o perquè, no havent-hi alternativa, ja eren suficients), no havíem previst a la línia d'ordres cap indicació que donés a l'usuari la confirmació que allò del *taijitu* que girava quan desplaçàvem el cursor era, efectivament, una invitació a activar caselles; després, en afegir el mètode 1 i el missatge *Sobre la VARIANT ESCOLLIDA (dreta), feu clic en les caselles que hagi de veure el jugador (per anul·lar els últims clics, polseu la tecla "<---- Backspace"):*, no ens vam adonar que el mètode 2 quedava coix, perquè ens limitàvem a fer desaparèixer de les tres línies d'ordres el final d'**EXPLICACIO**. I ara, amb la necessitat de donar cobertura als dos mètodes nous, la omissió s'ha fet palesa i mantindrem en la finestra de text el següents missatges fixos (tret del mètode 4, en què només servirà per donar entrada a la primera desactivació, es mantindran fixos fins a la consecució del SUDOKU-PROBLEMA -1- o fins a donar pas a informació sobre el nombre de solucions compatibles -2 i 3-):

1) *Sobre la VARIANT ESCOLLIDA (dreta), feu clic en les caselles que hagi de veure el jugador (per anul·lar els últims clics, polseu la tecla "<---Backspace"):*
2) *Podeu fer clic a qualsevol casella, sobre la finestra dreta o sobre l'esquerra (es passa d'una a l'altra amb un clic previ):*

3 i 4) *Podeu fer clic a qualsevol casella, sobre la VARIANT ESCOLLIDA (dreta):*

Us recordem que, malgrat que en l'últim missatge ja hem reflectit la limitació del joc a una finestra en els nous mètodes 3 i 4, perquè era molt fàcil d'aconseguir, encara no hem donat el pas a la restricció real. Però no només perquè duia força més feina i abans hem preferit avaluar la seva eficiència, sobretot la del mètode 3 (no fos cas que ens prenguéssim massa molèsties per depurar uns instruments que finalment es descarten), sinó perquè en un primer moment encara havíem complicat més el codi en la línia de seguir jugant amb les solucions normalitzades, en estar convençuts que malgrat tot seria necessari mentenir les dues etapes: en realitat, va ser l'avaluació experimental de l'eficiència del mètode 3 allò que ens va obrir els ulls, en constatar que (si l'encertàvem amb **NUM**) era innecessari fer escala a **PERFIL-V*V** + **COMPAT-PERFILS**, que podíem anar de dret a **ANORMALS** i que era millor.

Si en començar a activar caselles havíem detectat cert desequilibri explicatiu a favor del mètode 1, en ampliar la precisió final del mètode 2 (*EL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA. Omplint més caselles el fareu més fàcil (<Intro> o < > per acabar)*)

als mètodes 3 i 4, ens ha semblat que el mètode 1 es quedava orfe, en anunciar-se el SUDOKU-PROBLEMA exclusivament a l'àrea gràfica i passar directament la finestra de text a la postil·la *(Als efectes de revocació, l'execució de SUDÒKULUM compta com una sola ordre)*, comú a tots els mètodes. Com que aquest SUDOKU-PROBLEMA es caracteritza per no ser redundant, no tindria sentit utilitzar el mateix missatge final que la resta de mètodes, raó per la qual hem decidit que just abans de la postil·la aparegui el comiat *Ja teniu un SUDOKU-PROBLEMA amb les caselles justes!*.

Tot i que quasi tota la responsabilitat de control d'un trànsit que s'ha triplicat recau sobre la funció **FES-SUDOKU-234** (així és com hem rebateixat **FES-SUDOKU-2**), un mètode 3 encara llastat per la possibilitat de fer clic sobre la finestra esquerra (possibilitat real, tot i que la finestra de text digui *Podeu fer clic a qualsevol casella, sobre la VARIANT ESCOLLIDA (dreta):*) ens obligarà a determinats ajustos a **SUD-MIN** i **CLIC**, relacionats amb les noves variables **R**, **RX** i **RY**. Per justificar-les vindria bé recordar les explicacions del capítol 10 (*Solucions compatibles ho són totes les que hi són...*) respecte a un mètode 2 encara sense nom, i de **Q**, **QX** i **QY**. Dèiem allà que, quan fem clic a la finestra esquerra, l'obtenció del valor **N**, la determinació de **V** (la casella està activada i cal desactivar-la, o a l'inrevés), l'actualització de ****V*V**** (canviar **N** per coordenades o a l'inrevés) i la formació del conjunt de selecció **SS** per anar a **V+-** i visualitzar aquests canvis, s'apliquen primer en aquesta finestra; després, mitjançant **N->V**, trobem els valors **P**, **PX** i **PY** corresponents a la finestra dreta, actualitzem ****V**V**** i formem **SS** per anar a **V+-** i visualitzar els canvis; però, com que els valors exteriors a la funció **CLIC** han de ser els **P**, **PX** i **PY** corresponents a la primera finestra, abans assignem aquests valors a les variables de transició **Q**, **QX** i **QY**, per poder-los recuperar al final. Per contra, quan fem clic a la finestra dreta, el procediment s'aplica primer en aquesta finestra; després, mitjançant **V->N**, trobem els **P**, **PX** i **PY** corresponents a la finestra esquerra i hi apliquem el procediment, sense necessitat de **Q**, **QX** i **QY**. I afegíem que, quan a **FES-PUBLIC** (més endavant **FES-SUDOKU-2** i ara **FES-SUDOKU-234**) fèiem clic sobre una casella que formava part de **FOTO1** o **FOTO2**, se'ns demanava que confirméssim la desactivació i, si ens fèiem enrera, calia reactivar la casella desactivada, cosa que podíem aconseguir repetint l'accés a **CLIC**. Ara bé, segons la finestra on haguéssim fet clic abans del primer accés, havíem d'introduir canvis: si era a la finestra esquerra no hi havia cap problema, perquè tot estava preparat perquè els valors **P**, **PX** i **PY** finals corresponguessin a aquesta finestra; però si era a la de la dreta tot se'ns anava en orris, perquè els valors finals seguien corresponent a la finestra esquerra i ens calia recuperar els de la dreta. Per no haver de recórrer un altre cop a **V->N**, ens en sortíem adoptant també en els clics sobre la finestra dreta les variables de transició **Q**, **QX** i **QY**, però llavors calia diferenciar aquest segon accés a **CLIC** en la mateixa iteració **while** de **FES-PUBLIC** (en la versió d'ara hauríem de parlar del mateix accés a **FES-SUDOKU-234**) de forma inequívoca: a **FES-PUBLIC**, just abans del primer accés a **CLIC**, i en acabar **CLIC**, quan el clic s'hagués efectuat a la finestra esquerra, posaríem **Q** a **nil**; així **Q** només seria **T** en el segon accés i quan el clic s'hagués fet a la finestra dreta. Aquest dispositiu era tan embolicat precisament perquè les situacions eren simples (els dos **CLIC**s feien referència al mateix clic, i l'usuari no tenia l'oportunitat de canviar de finestra) i tractàvem de resoldre-les aprofitant recursos existents. Ara però, paradoxalment, si amb el nou mètode 3 aconseguim que **POST** s'activi quan només queda una solució compatible i entrem en una dinàmica descendent, ens trobem amb una situació més imprevisible i no hi haurà altre sortida que comptar amb més recursos, per bé que el dispositiu serà força més senzill. Per què la situació és més complicada?: perquè, mentre sigui **POST** = **nil**, l'usuari pot canviar de finestra a cada casella (provisionalment, repetim-ho un cop més) però, quan aquest senyal s'activi i de l'accés a **ANORMALS** en surti una única solució compatible, a l'hora de recular sobre els seus passos només disposarà del teclat i la finestra actual seguirà sent aquella des d'on havia efectuat l'últim clic (el **CLIC** desencadenant). En definitiva, ens trobem en una situació semblant a la descrita en el capítol 10 i que recordàvem fa un moment (semblant en el fet que els **CLIC**s que segueixen al primer -l'únic precedit per un autèntic clic interactiu- s'executen mantenint la mateixa finestra), però amb una diferència: que, quan rectificàvem la desactivació d'una casella enregistrada a **FOTO1** o **FOTO2**, el segon accés a **CLIC** en reproduïa un altre fet des d'aquesta mateixa finestra, mentre que, quan desfem el camí amb el mètode 3, el tercer, quart i següents poden reproduir **CLIC**s executats a l'ombra de l'altra finestra. És per això que la llista **PPM** que obrim en **SUD-MIN** quan **I = 17** no estarà composta per coordenades de caselles, com passava en el mètode 1, sinó per parells de coordenades de caselles: les homòlogues en ambdues finestres. Així, per a quan toqui desactivar una casella activada des de la finestra esquerra, sent la finestra dreta l'actual en el moment de les desactivacions, caldrà recollir-ne

les coordenades en aquesta segona (en la funció **CLIC**, mitjançant les variables **R**, **RX** i **RY**), per tal que **SUD-MIN** les emmagatzemi a **PPM** (junt amb **P**, **PX** i **PY**) i les puguem recuperar en la iteració **while** corresponent, en el dispositiu *ad hoc* muntat a **FES-SUDOKU-234** (no us ha de confondre l'aprofitament de la variable obsoleta **Q**).

I, en relació a **FES-SUDOKU-234**, només afegirem que l'esmentat bucle del mètode 3, que representa el replegament final fins a localitzar la casella que completa el SUDOKU-PROBLEMA, és local i no s'ha de confondre amb el del mètode 4, de dinàmica íntegrament descendent (tret d'episòdiques seqüències de reactivació de caselles desactivades), del qual cada accés a aquesta funció en constitueix una iteració: una cosa és el motor **while** ubicat a **FES-SUDOKU** i comú als quatre mètodes, inclòs el 3 pel que fa a l'etapa inicial **POST** = **nil**, de dinàmica sempre ascendent (tret d'episòdiques seqüències de desactivació de caselles activades), que s'ampliarà a a la segona si hem fet curt amb **NUM**, i un altre el **while** situat a **FES-SUDOKU-234** i mitjançant el qual retrocedirem, si **NUM** ha fet diana o ha sobrepassat l'objectiu. Per tal d'apreciar com ens ho hem fet per imbricar aquests mètodes en l'estructura ja prou complicada de **FES-SUDOKU-2**, hem subratllat les modificacions i additaments en el codi, i el mateix hem fet a la resta de funcions, tret de la nova **RETOL-234**.

```
(defun TRIA-METODE ()
  (textscr)
  (prompt "\n\n\n\n\nDisposeu de dos mètodes per obtenir un SUDOKU-PROBLEMA ")
  (prompt "d'una SUDOKU-SOLUCIÓ\n(aquesta VARIANT ESCOLLIDA) reactivant caselles")
  (prompt " (deixant-les visibles):")
  (prompt "\n\n1) Anar-ho fent fins haver activat tota la solució, moment en què")
  (prompt " el programa\n   remuntarà el procés per només deixar visibles les ")
  (prompt "caselles imprescindibles.")
  (prompt "\n\n2) Anar-ho fent fins que el programa ens informi que només hi ha ")
  (prompt "una solució\n   compatible amb les caselles activades ")
  (prompt "(SUDOKU-PROBLEMA): la SUDOKU-SOLUCIÓ\n   (entremig del procés es ")
  (prompt "produiran dues pauses, que poden ser ben llargues,\n   però tret ")
  (prompt "d'això l'activació de cada casella -revisable- serà immediata).")
  (initget "1 2 3 4")
  (setq 1-2 (getkword "\n\nTrieu l'opció 1, 2, 3 o 4 <3>: ") 1-2 (if 1-2 1-2 "3"))
  (if (= 1-2 "3")
      (progn
        (while (or (< (setq K (getint (strcat "\n¿Quantes caselles (entre 18 i 54"
                                              ") voleu emplenar, pel cap baix? <"
                                              (itoa NUM) ">: "))
                      K (if K K NUM)) 18)
                   (> K 54))
          (prompt (strcat "\n   El nombre de caselles no pot ser " (itoa K)
                          ": repetiu!")))
        (setq NUM K))))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if (and (> 1-2 "1") Q) (setq P Q X (car P) Y (cadr P) PX QX PY QY))
        (setq N (ELEMENT **9**9**)
              V (atom (ELEMENT **V**V**))
              **V**V** (if (and (= 1-2 "1") (not V)) **V**V** (COMMUTA **V**V**))
              SS (ssget "_C" PX PY))
        (if (and (> 1-2 "1") (not Q)) (setq Q P QX PX QY PY))
        (V+- "VAR-1")
        (if (> 1-2 "1")
            (progn
              (command "CVPORT" 2)
              (V->N)
              (setq **V*V** (COMMUTA **V*V**)
                    SS (ssget "_C" PX PY))
              (V+- "NORM-1")
              (command "CVPORT" 3))))
      (progn
        (setq N (ELEMENT **9*9**)
              V (atom (ELEMENT **V*V**))
              **V*V** (COMMUTA **V*V**)
              SS (ssget "_C" PX PY) Q P QX PX QY PY)
```

```
        (V+- "NORM-1")
        (command "CVPORT" 3)
        (N->V)
        (setq **V**V** (COMMUTA **V**V**)
              SS (ssget "_C" PX PY) R P RX PX RY PY)
        (V+- "VAR-1")
        (command "CVPORT" 2)
        (setq P Q X (car P) Y (cadr P) PX QX PY QY Q ()))))

(defun SUD-MIN (/ J0 J1 J2 J3 NOPOST)
  (setq I 0 J 0 K 0 POST T)
  (foreach LL (list **V*V** (TRANSPOSAR **V*V**))
    (if (and J1 POST)
        (if (= 1-2 "2")
            (if (< I 17) (setq POST ()))
            (progn
              (if (> I 16)
                  (setq PPM (cons (list (list P PX PY)
                                        (if (< (getvar "CVPORT") 3)
                                            (list R RX RY)
                                            (list Q QX QY)))
                                  PPM)))
              (if (< I NUM) (setq NOPOST T)))))
    (setq J3 1)
    (if POST
        (foreach L LL
          (if POST
              (progn
                (foreach E L
                  (if (atom E)
                      (progn
                        (if (or (and (not J1) (= J3 1))
                                (and J1 (not J2) (> J3 1)))
                            (setq I (1+ I)))
                        (setq J (1+ J)))))
                (setq J0 (cons J J0))
                (if (< K 2)
                    (setq K (1+ K))
                    (if (or (< J 2)
                            (equal J0 (list J 0 0))
                            (equal J0 (list J J 0))
                            (equal J0 (list J J J)))
                        (setq POST ())
                        (progn
                          (if J1
                              (if J2
                                  (setq J2 (if (< J J2) J J2)
                                        POST (or (< J3 3) (>= (+ J1 J2) 7)))
                                  (if (= J3 1)
                                      (setq J2 J)
                                      (setq J1 (if (< J J1) J J1))))
                              (setq J1 J))
                          (setq J0 () J 0 K 0 J3 (1+ J3)))))))))))
  (if (= 1-2 "2")
      (if (and POST (= I 17) (= (min J1 J2) 2)) (setq POST ()))
      (if (and POST NOPOST) (setq POST ()))))

(defun RE+MEMBER (R-9*9 R-LLL / 2R R-N R-P F C0 C3 C6)
  (if (or 1-2<34 (< (length KKK) 2))
      (if (COMPLET-9*9 R-9*9)
          (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9)
                                  SUDOKUS-V*V)
                KKK (cons T KKK))
          (progn
            (if (and PRE (> (cadr R-P) 0))
                (setq PRE () F (nth 0 R-9*9)
                      C0 (nth 0 F) C3 (nth 3 F) C6 (nth 6 F)
```

```
                    NORM (and (< C0 C3 C6)
                              (< C0 (nth 1 F) (nth 2 F))
                              (< C3 (nth 4 F) (nth 5 F))
                              (< C6 (nth 7 F) (nth 8 F)))))
          (if (or PRE (not NORM))
              (progn
                 (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
                (foreach E R-N
                  (if E (progn
                          (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                          (RE+MEMBER (car 2R) (cadr 2R))))))
              (if NORM (setq PRE T))))))

(defun ANORMALS (NORM=1 1-2<34 / L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P PRE
                                 NORM)
  (setq L0A8 '(0 1 2 3 4 5 6 7 8) FOTO1 () FOTO2 (FOTO-V*V))
  ......................................
  (if (and NORM=1 (= KK 1)) (setq KKK ())))

(defun RETOL-234 ()
  (RETOL-2 "Determinar" "quantes" "solucions" "compatibles"
           "hi ha" "pot trigar" "una bona" "estona." "Espereu..."))

(defun FES-SUDOKU-234 (/ LINIES-V*V KLINIES-V*V *K* *KK* R RX RY) ; Substitueix
  (setq Q ())                                                     ; FES-SUDOKU-2
  (CLIC)
  (if POST
      (if V
          (if (or (member P FOTO1) (member P FOTO2))
              (if (= 1-2 "4")
                  (setq POST ())
                  (progn
                    (command "ESPACIOP" "BORRA" "LT" ""
                             "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                    (RETOL-2 "Anul-lant" "aquesta" "casella," "caldrà"
                             "recalcular" "solucions" "compatibles"
                             "i tornar a" "esperar...")
                    (if FOTO2
                        (progn
                          (SEGUIR T)
                          (command "DESHACER" "R" "DESHACER" "M"
                                   "INSERT" "C" "0,0" "" "" "")
                          (RETOL-2 "...i, si surt" "\"EL SUDOKU" "JA TÉ UNA"
                                   "SOLUCIÓ" "ÚNICA\", mai" "no sabreu"
                                   "si sobren" "caselles" "plenes.")))
                    (initget "Si No")
                    (setq CONF
                          (getkword "\n.\n.\nConfirmeu l'anul·lació (S/N)? <N>: ")
                          CONF (if CONF CONF "No"))
                    (if (= CONF "Si")
                        (setq POST ())
                        (progn
                          (command "DESHACER" "R" "UY" "ESPACIOM")
                          (CLIC))))))
              (progn
                (setq I (assoc P S-V*V) S-V*V (SUPR I S-V*V)
                                  ; És indiferent usar **9*9** o **V*V**.
                      I (caar S-V*V) J (nth (car I) (nth (cadr I) **V*V**)))
                (NUMCOMPAT I J ()))))
          ; En les 6 línies següents, el codi subratllat únicament té el propòsit
          ; d'advertir l'usuari del mètode 3 sobre la possibilitat que l'execució
          ; de NUMCOMPAT no sigui tan ràpida (pràcticament instantània) com en el
          ; mètode 2. Si un valor massa curt de NUM provoqués pauses sensibles en
          ; l'emplenament de caselles posterior a l'activació de POST, l'aparició
          ; d'una nota aclaratòria tranquil·litzarà l'usuari.
          (progn
            (command "ESPACIOP" "DESHACER" "M"
                     "INSERT" "C" "0,0" "" "" "")
```

```
          (RETOL-234)
          (NUMCOMPAT P N T)
          (command "DESHACER" "R"))))
(if (not POST)
    (progn
      (if (and V (or (member P FOTO1) (member P FOTO2)))
          (setq POST T)
          (if (not V) (SUD-MIN)))
      (if POST
          (progn
            (if V
                (if (< 1-2 "4") (command "DESHACER" "R") (command "ESPACIOP"))
                (progn
                  (command "ESPACIOP")
                  (if (= 1-2 "2") (command "BORRA" "LT" ""))))
            (if (or (not (or FOTO1 FOTO2))
                    (member P FOTO1))
                (if (= 1-2 "2")
                    (progn
                      (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                      (RETOL-2 "Determinar" "quantes" "solucions" "normalitza-"
                               "des són" "compatibles" "pot trigar" "hores."
                               "Espereu...")
                      (PERFIL-V*V)
                      (COMPAT-PERFILS)
                      (command "DESHACER" "R" "UY" "ESPACIOM"))
                    (progn
                      (prompt "\n.\n.\n")
                      (command "INSERT" "C" "0,0" "" "" ""
                               "DESHACER" "M")
                      (RETOL-234)
                      (ANORMALS () T)
                      (command "DESHACER" "R")
                      (if (= KK 1)
                          (progn
                            (while
                              (and (= KK 1) (> (length PPM) 1))
                              (prompt "Només hi ha una solució compatible i, ")
                              (prompt "just per això, cal mirar\nquantes n'hi")
                              (prompt " hauria amb una casella menys (")
                              (prompt (itoa (setq NUM (1- NUM))))
                              (prompt " caselles plenes).")
                              (SEGUIR ())
                              (command "ESPACIOM")
                              (setq Q (car PPM) PPM (cdr PPM)
                                    Q (if (< (getvar "CVPORT") 3)
                                          (car Q)
                                          (cadr Q))
                                    P (car Q) X (car P) Y (cadr P)
                                    PX (cadr Q) PY (last Q) Q ())
                              (CLIC)
                              (command "ESPACIOP")
                              (if (> (length PPM) 0)
                                  (progn
                                    (prompt "\n.\n.\n")
                                    (command "DESHACER" "M")
                                    (RETOL-234)
                                    (ANORMALS () ())
                                    (command "DESHACER" "R"))))
                            (if (> KK 1)
                                (progn
                                  (prompt "Us heu passat de la ratlla: ")
                                  (prompt "amb més d'una solució compatible, ")
                                  (prompt "el sudoku\nhaurà de recuperar ")
                                  (prompt "l'última casella esborrada ("
                                  (prompt (itoa (setq NUM (1+ NUM))))
                                  (prompt " caselles plenes).")
                                  (SEGUIR ())
```

```
                                    (command "ESPACIOM")
                                    (CLIC)
                                    (command "ESPACIOP")
                                    (NUMCOMPAT P N T))
                                  (prompt (strcat "\n.\nHeu tingut el privilegi "
                                                  "de crear un SUDOKU MÍNIM "
                                                  "(només 17 caselles plenes)."))))))
                        (command "BORRA" "LT" "")))
                  (progn
                    (command "DESHACER" "M"
                             "INSERT" "C" "0,0" "" "" "")
                    (RETOL-234)
                    (ANORMALS () (< 1-2 "4"))
                    (command "DESHACER" "R")
                    (if (< 1-2 "4") (command "UY" "ESPACIOM"))))))))
(if POST
    (progn
      (setq GR (strcat "\nEL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA.\nOmplint més case"
                       "lles el fareu més fàcil (<Intro> o < > per acabar)"))
      (if (or (and (= 1-2 "4") (= KK 1) (> (setq NUM (length FOTO2)) 17))
              (and (< 1-2 "4") (> KK 1)))
          (if (= 1-2 "4")
              (prompt (strcat "\n.\nAmb només una solució compatible, heu de "
                              "buidar més caselles:"))
              (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                              (if KKK "" "normalitzades ")
                              "heu d'omplir més caselles:")))
          (if FOTO2
              (progn
                (if (= 1-2 "4")
                    (progn
                      (if (> KK 1)
                          (progn
                            (command "ESPACIOP" "DESHACER" "M"
                                     "INSERT" "C" "0,0" "" "" "")
                            (prompt "\n.\nUs heu passat de la ratlla: amb més ")
                            (prompt "d'una solució compatible, el sudoku\n")
                            (prompt "haurà de recuperar l'última casella ")
                            (prompt (strcat "esborrada (" (itoa NUM)))
                            (prompt " caselles plenes).")
                            (SEGUIR ())
                            (command "DESHACER" "R" "ESPACIOM")
                            (CLIC)
                            (command "ESPACIOP" "DESHACER" "M"
                                     "INSERT" "C" "0,0" "" "" "")
                            (NUMCOMPAT P N T)
                            (command "DESHACER" "R"))
                          (prompt "\n.\nHeu tingut el privilegi de crear un "
                                  "SUDOKU MÍNIM (només 17 caselles plenes)."))
                      (setq 1-2 "3")))
                (if KKK (setq FOTO2 (FOTO-V*V) KKK ()))
                (prompt (strcat (if (and (> 1-2 "2") (= NUM 17))
                                    (progn (setq PPM () NUM ()) "")
                                    "\n.")
                                "\n" GR)))
              (progn
                (foreach E '(T ())
                  (prompt (strcat "\n.\nTeniu 1 solució normalitzada compatible"
                                  " amb V. E. NORMALITZADA."))
                  (if E (SEGUIR ()) (terpri)))
                (command "ESPACIOP"
                         "DESHACER" "M"
                         "INSERT" "C" "0,0" "" "" "")
                (RETOL-2 "Però pot" "haver-n'hi" "més, de no" "normalitza-"
                         "des però" "igualment" "compatibles." "Així que"
                         "espereu...")
                (ANORMALS T T)
                (command "DESHACER" "R" "UY" "ESPACIOM")
```

```
                (if (> KK 1)
                    (progn
                      (if (> KK 2)
                          (prompt (strcat "\n.\nHi ha " (itoa (1- KK))
                                          " solucions més, no normalitzades"
                                          " però també compatibles,"))
                          (prompt (strcat "\n.\nHi ha una solució més, no nor"
                                          "malitzada però també compatible,")))
                      (prompt "\naixí que heu d'omplir més caselles:"))
                    (prompt (strcat "\n.\nCom no n'hi ha cap de "
                                    "no normalitzada, " GR)))))))))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     **V**V**)
  (if (= 1-2 "1")
      (progn
        (if (not LLISTES) (INI-LLISTES))
        (INI-LLL ())
        (setq NIL*NIL LLL 0*0 (INI-COORDS) POST T)
        (INI-LLL L1A9))
      (progn (NORMTEXT-9*9) (EXPLICACIO)))
  (setq PPM () **V**V** (if (< 1-2 "4") (INI-COORDS) **9**9**)
        **V*V** (if (> 1-2 "1")
                    (if (< 1-2 "4") **V**V** **9*9**))
        K 1
        GR (if (= 1-2 "1")
               (strcat "Sobre la VARIANT ESCOLLIDA (dreta), feu clic en les "
                       "caselles que hagi de veure\nel jugador (per anul•lar "
                       "els últims clics, polseu la tecla \"<---Backspace\"):")
               (strcat "Podeu fer clic a qualsevol casella, sobre la "
                       (if (= 1-2 "2")
                           (strcat "finestra dreta o sobre l'esquerra\n(es "
                                   "passa d'una a l'altra amb un clic previ):")
                           "VARIANT ESCOLLIDA (dreta):"))))
  (prompt (strcat "\n.\n.\n" GR))
  (if (= 1-2 "4") (setq FOTO2 (FOTO-V*V) POST T))
  (repeat 2
    (setq K (1+ K) J (if (= K 2) "NORM-0" "VAR-0"))
    (if (or (> K 2) (not (and (= ABC "A") (= 1-2 "1"))))
        (progn
          (command "ESPACIOM" "CVPORT" K)
          (if (= 1-2 "4")
              (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
                       "CAMBIA" "P" "" "P" "L" 1 ""
                       "BORRA" "P" ""))
          (command "CAMBIA" "C" "0,0" "8,8" "" "P" "C" J "")
          (if (= 1-2 "4") (command "UY" "ORDENAOBJETOS" "P" "" "D")))))
  (graphscr)
  (YIN-YANG)
  (while (not (or (and (= 1-2 "1") (COMPLET-9*9 **V**V**))
                  (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if (or (= 1-2 "2") (= (getvar "CVPORT") 3))
             (if (= (car GR) 5)
                 (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
                 (if (or (and (= 1-2 "1") PPM (equal GR '(2 8)))
                         (and (= (car GR) 3)
                              (> (getvar "CVPORT") 1)
                              (setq P (cadr GR) PX (car P) PY (cadr P))
                              (equal (list PX PY) '(4 4) 4.48)
                              (or (< (- PX (setq X (fix PX))) 0.48)
                                  (< (- (setq X (fix (1+ PX))) PX) 0.48))
                              (or (< (- PY (setq Y (fix PY))) 0.48)
                                  (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                              (or (= 1-2 "2") (listp (ELEMENT **V**V**)))
                              (PUNT)))
                     (if (= 1-2 "1") (FES-SUDOKU-1) (FES-SUDOKU-234))))))
```

```
  (if (= 1-2 "1")
      (progn
        (while (< (last (car PPM)) 2) (setq PPM (cdr PPM)))
        (setq SS ())
        (foreach E PPM (if (> (last E) 0) (setq SS (cons E SS))))
        (setq PPM SS SS (ssadd))
        (RASTRE))
      (command "ESPACIOP"))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" ""))

(defun C:SUDOKULUM (/ ANG TG NUM L1A9 LLISTES CONF ABC 1-2 0*0 3*N NIL*NIL G I J
                      K L LL LLL M N N1 N2 N3 N4 P PPM PX PY Q QX QY SS V X Y OK
                      GR POST OSN FIL SRT SVT ECO TN/R-L TT **9*9** **9**9**
;                     HPD DRA LAY                          ; Només si és ACAD 2011.
                   )
  (setq ANG -0.5
        TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9)              ; (funció SUD-MIN).
        NUM 36)  ; Valor per defecte en nombre de caselles plenes que activa POST
  ..................................................
  (if (/= ABC "D") (progn (TRIA-METODE) (FES-SUDOKU)))
  (prompt "\n.")
  (if (= 1-2 "1")
      (prompt "\nJa teniu un SUDOKU-PROBLEMA amb les caselles justes!"))
  (prompt "\n(Als efectes de revocació, l'execució de SUDÒKULUM ")
  (prompt "compta com una sola ordre.)")
  (command "OSMODE" OSN
           "FILLMODE" FIL
           "SORTENTS" SRT
           "SAVETIME" SVT
;          "HPDRAWORDER" HPD                              ; Només si és ACAD 2011.
;          "DRAWORDERCTL" DRA                             ; Només si és ACAD 2011.
;          "LAYLOCKFADECTL" LAY                           ; Només si és ACAD 2011.
           "DESHACER" "F")
  (setvar "CMDECHO" ECO)
  (princ))
```

En relació a certs malabarismes de la funció **FES-SUDOKU** associats al mètode 4, que reproduirem aquí per evitar-li al lector anar repetidament a l'última pàgina i per desempallegar el codi de subratllats, convé que anem per pams:

```
  .........
  (repeat 2
    (setq K (1+ K) J (if (= K 2) "NORM-0" "VAR-0"))
    (if (or (> K 2) (not (and (= ABC "A") (= 1-2 "1"))))
        (progn
          (command "ESPACIOM" "CVPORT" K)
          (if (= 1-2 "4")
              (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
                       "CAMBIA" "P" "" "P" "L" 1 ""
                       "BORRA" "P" ""))
          (command "CAMBIA" "C" "0,0" "8,8" "" "P" "C" J "")
          (if (= 1-2 "4") (command "UY" "ORDENAOBJETOS" "P" "" "D")))))
          ..........................................................
```

D'una banda hem de recordar que, des de **GRAF-9*9** < **NORMTEXT-9*9** < **FES-SUDOKU**, les solucions que veiem a dreta (VARIANT ESCOLLIDA) i esquerra (V. E. NORMALITZADA) es representen en blanc, per estar situades en les capes NORM-1 i VAR-1, però que amb els mètodes 1 i 2 (i ara també el 3) cal partir de les caselles en gris (NORM-0 i VAR-0), per anar-hi sumant les activacions en blanc (NORM-1 i VAR-1), raó per la qual recorríem a **CAMBIA** per traslladar-les en bloc des d'unes capes a les altres. Però el mètode 4 arrenca de l'escaquer 9×9 completament activat i el primer clic sobre cada casella en comportarà la desactivació, així que la situació de partença ha de ser diferent: cal que totes les caselles estiguin emplenades doblement, amb el caràcter numèric corresponent situat a la capa NORM-0 o VAR-0 (en gris), però també repetit a la capa NORM-1 o VAR-1 (en blanc) i amb aquesta última presència (que, com vam quedar de fer en el capítol *Assegurar-se una visualització correcta*, elevarem fins a la cota **z = 1**) predominant visualment sobre l'anterior. La manera més senzilla d'assolir-ho és copiar el contingut de NORM-1 i VAR-1, passar un dels exemplars a **z = 1** i esborrar-lo, traslladar l'altre a NORM-0 i VAR-0, i recuperar

el primer mitjançant l'ordre **UY**. Hem tret partit del dispositiu de selecció **Previo** però encara en treurem més. Treballant amb ACAD 2011, si únicament haguéssim fet

```
(if (= 1-2 "4")
    (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
             "CAMBIA" "P" "" "P" "L" 1 ""
             "BORRA" "P" ""))
(command "CAMBIA" "C" "0,0" "8,8" "" "P" "C" J "")
(if (= 1-2 "4") (command "UY"))
```

tant a la finestra esquerra (V. E. NORMALITZADA) com a la dreta (VERSIÓ ESCOLLIDA) d'entrada prevaldria visualment el contingut de les capes NORM-1 i VAR-1, però no passarien de blanc a gris (capes NORM-0 o VAR-0) en desactivar-les. És més: si en reactivéssim alguna, la resta de caselles activades passarien a veure's en gris. Però no n'hi haurà prou a aplicar **ORDENAOBJETOS Delante** als objectes recuperats, perquè algunes caselles encara poden aparèixer en gris (aquestes deficiències no s'esdevenen sempre, ni tan sols amb una mateixa VERSIÓ ESCOLLIDA); per evitar-ho, **PREPGRAF-9*9** (primera funció reproduïda) haurà de posar a **0** la variable de sistema **DRAWORDERCTL** i, en acabar l'execució, **C:SUDOKULUM** restaurar-la al seu valor previ. Copsarem millor l'efecte dels elements en joc prescindint del fet que la variable de sistema **DRAWORDERCTL** ja l'havíem posada a **0** el capítol precedent (*Assegurar-se una visualització correcta*) i els anem aplicant sistemàticament:

- Sense tocar **DRAWORDERCTL** (el valor per defecte és **1**) i sense recórrer a l'ordre **ORDENAOBJETOS**, amb ACAD 2011 les dues finestres apareixerien amb tots els valors en gris, i passarien a blanc a mesura que les anéssim desactivant (justament al revés de com volem), aspecte que no variaria reactivant-les amb un altre clic.
- Passant **DRAWORDERCTL** a **0** però encara sense recórrer a **ORDENAOBJETOS**, les dues finestres apareixerien amb tots els valors en blanc, i les caselles desactivades seguirien en blanc. Però si reactivem alguna casella desactivada, totes les que no haguem tocat passen a gris, i a partir d'aquest moment tot funcionarà com en el cas precedent.
- Amb **DRAWORDERCTL** a **0** i fent **ORDENAOBJETOS D**, tot anirà com una seda: les dues finestres apareixeran amb tots els valors en blanc, les caselles desactivades passaran a gris i, si les reactivem, tornaran a veure's en blanc sense alterar l'aspecte de la resta de caselles.

I a ACAD 2004 (tant si és **MODOSOMBRA 2D** com **O**culto) també li cal **ORDENAOBJETOS D**.

Només afegirem una observació, a propòsit de l'estranyesa que pot provocar el fet que ens haguem molestat a convertir en funció una aplicació particular de **RETOL-2** (els nou textos que componen el missatge *Determinar quantes solucions compatibles hi ha pot trigar una bona estona. Espereu...*), repetida 4 cops en **FES-SUDOKU-234** (que n'haurien pogut ser 5, si no haguéssim volgut dramatitzar més l'advertiment per a l'execució de **PERFIL-V*V** i **COMPAT-PERFILS**, afegint-hi *normalitzades* i posant *hores* en comptes d'*una bona estona*), i que tanmateix no haguem fet cap funció com

```
(defun AVIS ()
  (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
  (RETOL-234))
```

per simplificar el codi, atès que aquestes ordres d'AutoCAD apareixen força cops en aquesta mateixa funció. S'hauria pogut fer, però al final vam optar per deixar-ho córrer, perquè només hi ha dos fragments en què **AVIS** s'hauria pogut aplicar:

1)
```
......
(progn
  (command "ESPACIOP" "DESHACER" "M"
           "INSERT" "C" "0,0" "" "" "")
  (RETOL-234)
  (NUMCOMPAT P N T)
  (command "DESHACER" "R"))))
```

hauria pogut quedar així

```
......
(progn
  (command "ESPACIOP")
  (AVIS)
  (NUMCOMPAT P N T)
  (command "DESHACER" "R"))))
```

i 2)
```
      ......
      (progn
        (command "DESHACER" "M"
                 "INSERT" "C" "0,0" "" "" "")
        (RETOL-234)
        (ANORMALS () (< 1-2 "4"))
```

```
                    (command "DESHACER" "R")
                    (if (< 1-2 "4") (command "UY" "ESPACIOM"))))))))
```

hauria pogut quedar així

```
                  ......
                  (progn
                    (AVIS)
                    (ANORMALS () (< 1-2 "4"))
                    (command "DESHACER" "R")
                    (if (< 1-2 "4") (command "UY" "ESPACIOM"))))))))
```

Però en els altres dos fragments, que pertoquen als mètodes 3 (els dos) i 4 (només el primer i la línia final),

```
                      ......
                      (progn
                        (prompt "\n.\n.\n")
                        (command "INSERT" "C" "0,0" "" "" ""
                                 "DESHACER" "M")
                        (RETOL-234)
                        (ANORMALS () T)
                        (command "DESHACER" "R")
                        (if (= KK 1)
                            (progn
                              (while (and (= KK 1) (> (length PPM) 1))
                                .....................
                                (if (> (length PPM) 0)
                                    (progn
                                      (prompt "\n.\n.\n")
                                      (command "DESHACER" "M")
                                      (RETOL-234)
                                      (ANORMALS () ())
                                      (command "DESHACER" "R")))) .......))
                        (command "BORRA" "LT" "")))
```

això ja no hauria estat possible, perquè en el primer (que correspon a l'activació de **POST** havent fet llarg, és a dir, trobant-nos amb una única solució compatible i veient-nos obligats a recular sobre els nostres passos, prement qualsevol tecla) l'acció ja és atípica: la marca **M** de **DESHACER** la posem abans d'accedir a **RETOL-234** (els nou textos *Determinar quantes solucions compatibles hi ha pot trigar una bona estona. Espereu...*) però després d'haver inserit la tapadora circular gris **"C"**, ja que, fins que no aïllem el SUDOKU-PROBLEMA sense redundància afegida i estiguem en disposició d'acabar o d'activar caselles suplementàries, en tindrem prou a polsar tecles i no necessitarem veure el símbol YIN-YANG de confirmació del moviment del cursor; cada cop que fem **DESHACER R** després d'executar **ANORMALS**, només s'esborrarà l'avís i recuperarem un neutre rectangle gris enllaçant les dues finestres, que en realitat serà el rectangle amb un forat circular (l'encaix del YIN-YANG) tapat pel bloc **"C"**. Desactivant caselles amb el mètode 4 (situació en que sí que caldrà fer CLIC discrecionalment sobre les que vulguem, en no haver-hi camí previ que poguem desfer) i també havent aïllat ja el SUDOKU-PROBLEMA amb els mètodes 3 i 4, l'ordre **BORRA LT** ens permetrà d'eliminar la tapadora **"C"** i deixar un cau per al YIN-YANG.

Ara que tenim les noves eines (tret dels aspectes formals de conducció de l'usuari i de certes incoherències residuals de què hem fet esment), ja és hora de mostrar la seva superioritat sobre el mètode 2. I, perquè ningú sospiti que ens treiem de la màniga uns exemples especialment seleccionats per causar admiració, recuperarem tres vells coneguts: el sudoku amb **23 caselles** de LA VANGUARDIA del dia 25-02-08 (utilitzat des del capítol 9), el que figurava amb **21 caselles** just després de les solucions PANÒNICA i CANÒNICA (capítol 6) i el primer de la col·lecció de sudokus de **17 caselles** publicada per Gordon Royle.

Amb el de 23 emplenaments recordarem que, dintre del segon annex (*Actualització a l'última versió de FES-SOLUCIO del capítol 6- Etapa prèvia: crear una solució, IV (... o a Camprodon)*) al capítol 12 (*Un nou camí i també una drecera*), aconseguíem abaixar les durades dels dos dispositius "maquinària pesant", seguint la seqüència d'emplenaments de sempre: el tàndem **PERFIL-V*V** + **COMPAT-PERFILS** (treballant amb 19 emplenaments) trigava **3'10"** a calcular que hi havia 1.117 solucions normalitzades compatibles, i **ANORMALS** (amb 22 emplenaments) trigava **0'20"** a dir que hi havia un total de 593 solucions compatibles. Doncs bé: fent-ho amb el nostre flamant mètode 3 (o amb el 4, si tenim la paciència d'anar buidant la VERSIÓ ESCOLLIDA), la marxa cap enrera (prenent el valor per defecte **NUM = 36**, o qualsevol **NUM ≥ 23**) serà una successió de passos "instantanis" (entenent per tal durades inferiors a un segon).

Amb el de 21 emplenaments, si la seqüència d'emplenaments consisteix a emplenar les caselles públiques dels requadres 9×3 inferior i superior (15 emplenaments, en total), abans d'atacar el del mig, i en aquest requadre començar per les tres de l'esquerra (1,6 (**6**), 2,6 (**4**) i 3,6 (**9**)) i seguir amb 7,5 (**1**), moment en què **POST** s'activarà, amb el mètode 2, **PERFIL-V*V** + **COMPAT-PERFILS** (19 emplenaments) trigarà **6'05"** a dir-nos que hi ha 218 solucions normalitzades compatibles, i **ANORMALS** (21 emplenaments) només triga un instant (en el sentit que abans hem atorgat al mot) a dir-nos que l'única solució normalitzada compatible també és l'única compatible. Enfront d'aquests resultats, la marxa cap enrera amb els mètodes 3 (amb **NUM ≥ 21**) o 4 serà una successió de passos instantanis, tret del penúltim (21 emplenaments), que trigarà **0'10"**. El guany és evident perquè hem estalviat 6 minuts, i no us ha d'estranyar que en casos així, amb emplenaments iguals, **ANORMALS** trigui menys amb el mètode 2 que amb el 3 o 4: en el primer cas l'argument binari **NORM=1** és **T** i en el segon **nil**; si repasseu el final del capítol 11 (*... Però no hi eren totes les que ho són*), veureu com aquest dispositiu li evitava a **RE+MEMBER** perdre el temps progressant en l'exploració de solucions normalitzades (detectables per la primera fila de la transposada) quan ja sabíem que, de compatibles, només n'hi havia una.

Amb el de 17 emplenaments el mètode 2 fracassa perquè l'ordinador s'acaba penjant. Davant d'això, cal acceptar els mètodes 3 (amb **NUM ≥ 17**) i 4 com un èxit relatiu, perquè malgrat les esperes ens permeten d'arribar al final. Per exemple, si prenem **NUM = 25** i, més enllà del SUDOKU-PROBLEMA, representat a l'esquerra, seguim amb l'activació de les caselles 18ª (4,9 (**7**)), 19ª (9,8 (**6**)), 20ª (6,7 (**3**)), 21ª (5,8 (**1**)), 22ª (9,1 (**9**)), 23ª (8,4 (**2**)), 24ª (1,5 (**5**)) i 25ª (2,9 (**9**)), incorporades a la prepresentació del mig, el missatge *Només hi ha una solució compatible i, just per això, cal mirar quantes n'hi hauria amb una casella menys (24 caselles plenes)* s'anirà repetint per a valors descendents, amb temps d'espera que aniran creixent:
- Després de desactivar la 25ª (*... (24 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 24ª (*... (23 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 23ª (*... (22 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 22ª (*... (21 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 21ª (*... (20 caselles plenes)*), caldrà esperar **0'10"**.
- Després de desactivar la 20ª (*... (19 caselles plenes)*), caldrà esperar **0'40"**.
- Després de desactivar la 19ª (*... (18 caselles plenes)*), caldrà esperar **6'45"**.
- Després de desactivar la 18ª (*... (17 caselles plenes)*), caldrà esperar **50'00"**.

Però, a diferència d'altres situacions, no caldrà desactivar la casella anterior, esperar el missatge *Us heu passat de la ratlla: amb més d'una solució compatible, el sudoku haurà de recuperar l'última casella esborrada (NN caselles plenes)* i polsar qualsevol tecla per recuperar l'última casella desactivada, perquè aquí el programa (sabedor que no hi ha sudokus de menys de 17 caselles) ens donarà la bona nova *Heu tingut el privilegi de crear un SUDOKU MÍNIM (només 17 caselles plenes)*.

6 9 3	7 8 4	5 **1** 2	6 **9** 3	**7** 8 4	5 **1** 2	9 1 **2**	3 4 5	6 7 8
4 8 7	5 1 2	9 3 6	**4** 8 7	5 **1** 2	9 3 **6**	6 7 8	9 1 2	**3 4 5**
1 **2** 5	9 6 3	8 7 4	1 **2** 5	9 6 **3**	8 7 4	3 4 5	**6 7 8**	9 1 2
9 3 2	6 **5** 1	**4** 8 **7**	9 3 2	6 **5** 1	**4** 8 **7**	8 9 1	2 3 4	5 6 7
5 6 **8**	2 4 7	**3** 9 1	**5** 6 **8**	2 4 7	**3** 9 1	5 6 7	8 9 1	**2 3 4**
7 4 **1**	3 **9** 8	6 2 5	7 4 **1**	3 **9** 8	6 **2** 5	2 3 4	**5 6 7**	8 9 1
3 1 9	**4** 7 5	**2** 6 8	**3** 1 9	**4** 7 5	**2** 6 8	7 8 9	1 2 3	4 5 6
8 **5** 6	**1** 2 9	7 4 3	8 **5** 6	**1** 2 9	7 4 3	4 5 6	**7 8 9**	1 2 3
2 7 4	**8** 3 **6**	1 5 9	2 7 4	**8** 3 **6**	1 5 **9**	**1 2 3**	4 5 6	7 8 9

I la representació de la dreta? Doncs, si us hi fixeu bé veureu que és la SOLUCIÓ CANÒNICA, amb què havíem estat fent proves i d'on, per pura xiripa, vam descobrir un SUDOKU PROBLEMA de **19 caselles**: el constituït per les tipografiades en negreta. Probablement pensareu que la troballa es va produir carregant-la amb l'opció C i jugant-hi amb el mètode 1, però no va anar ben bé així. Tant si us ho creieu com si no, va ser tractant d'obtenir un variat repertori de SUDOKUs PROBLEMA (variat pel que feia a nombre d'emplenaments) mitjançant l'opció A (SOLUCIÓ CREADA), quan l'autor anava emplenant caselles de manera que el seu contingut coincidís amb la SOLUCIÓ CANÒNICA (que tenia en paper, al costat de l'ordinador): la intenció era passar després a la fase de postsolució, per posar a prova el flamant mètode 3. I realment va anar bé disposar d'aquest nou sudoku, perquè amb ell retrobàvem una baula perduda: resignats a no poder completar el catàleg amb sudokus de 18, 20 i 22 caselles, ja era prou bo comptar amb un repertori de senars (17, **19**, 21 i 23,

perquè els més poblats no representaven cap problema). Pel que fa al mètode 2, hem avançat una mica respecte al sudoku mínim de 17: només una fórmula d'activació ens permetrà d'emplenar 18 caselles, abans que s'activi **POST**, i completar el procés; en totes les demés només s'arribarà a emplenar-ne 17 i l'ordinador acabarà penjat. La fórmula viable consisteix a començar amb els 17 emplenaments compresos entre la 3ª i la 9ª columna, i seguir amb la casella 2,1 (**2**): **PERFIL-V*V** + **COMPAT-PERFILS** trigarà **3'40"** a dir-nos que només hi ha una solució normalitzada compatible i, en aquesta mateixa casella, **ANORMALS** trigarà uns altres **3'40"** a dir-nos que n'hi ha 11.721 més, no normalitzades però també compatibles; en activar 1,1 (**1**), **NUMCOMPAT** ens informarà instantàniament que ja tenim un SUDOKU-PROBLEMA. Si alternativament utilitzem el mètode 3, un altre cop amb **NUM = 25**, i seguim amb l'activació de les caselles 20ª (7,3 (**4**)), 21ª (8,3 (**5**)), 22ª (9,3 (**6**)), 23ª (1,6 (**8**)), 24ª (2,6 (**9**)) i 25ª (3,6 (**1**)), per exemple, la reculada pas a pas es produirà d'aquesta manera:
- Després de desactivar la 25ª (*... (24 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 24ª (*... (23 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 23ª (*... (22 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 22ª (*... (21 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 21ª (*... (20 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 20ª (*... (19 caselles plenes)*), caldrà esperar **0'00"**.
- Després de desactivar la 19ª (*... (18 caselles plenes)*), caldrà esperar **0'00"**.

Tot seguit, sortirà el missatge *Us heu passat de la ratlla: amb més d'una solució compatible, el sudoku haurà de recuperar l'última casella esborrada (19 caselles plenes)*, i reactivant la 19ª casella tindrem el SUDOKU-PROBLEMA. Allà on el mètode 2 fracassava, els 3 i 4 també se'n surten, i amb respostes igualment instantànies.

Exposada aquesta millora substancial, quedava per veure la que havíem apuntat en primer lloc i que havíem presentat com una tècnica per millorar l'eficiència del mètode 2: transposar en determinats casos la pseudomatriu 9×9 de treball ****V*V****. Això, es clar, si al final decidim ser conservadors i optem per no llençar a les escombraries aquest mètode, clarament superat pel que acabem d'explicar, tot i que el poc cost de la transposició ens podria decidir a estendre-la a les situacions on la maquinària pesant, en comptes de **PERFIL-V*V** + **COMPAT-PERFILS**, fos **ANORMALS**. Malgrat això, és en relació al funcionament d'aquest tàndem (que, com el lector recordarà del capítol 8 i immediatament següents, inventaria files compatibles amb cadascuna de les files de la solució escollida, tríades de files compatibles entre si, adscrites a cada requadre 9×3, i combinacions de tríades compatibles entre si) que la millora en el rendiment d'aquestes funcions, treballant amb la pseudomatriu transposada quan l'ocupació dels requadres 9×3 és més uniforme que amb l'original, es pot justificar raonadament. (Amb **ANORMALS**, tanmateix, ni s'ha pogut constatar empíricament que millori el rendiment ni hem volgut entrar a estudiar seriosament les conseqüències que podria tenir la transposició, perquè dels 5 fragments del capítol *Etapa prèvia: crear una solució* ja n'havíem tret com a conclusió la gran complexitat del tema i la seva exagerada dependència de particularismes: allà eren les autorecursions de **RE-MEMBER** i aquí ho serien les de **RE+MEMBER**.) Paradoxalment, els pressupostos teòrics en què inicialment se sustentava la intuició de partença eren radicalment falsos, i el fet que el procediment en veiés àmpliament recolzat pels resultats experimentals va ser allò que va contribuir a no dissipar l'error fins pràcticament arribat el moment de plasmar l'argumentació en aquestes línies.

Va ser el sempitern sudoku difícil de LA VANGUARDIA (25-02-08) el que va desfermar aquest embolic. I no precisament l'aproximació que fèiem al llarg dels capítols 9 a 12 i que hem reprès fa poc per verificar els guanys obtinguts apujant el llistó a **SUD-MIN**, sinó l'intent de avançar l'activació de **POST**, amb una casella emplenada de menys en relació a la seqüència d'emplenaments que havíem vist: tot consistia a saltar-se l'emplenament de la casella 8,3 (**6**), i així **SUD-MIN** activaria **POST** quan l'emplenament de 4,7 (**3**) trenqués el condicionament de quedar algun requadre 3×9 amb dues columnes buides, amb només 18 caselles ocupades en lloc de 19. Per tal de facilitar el seguiment del que explicarem, reproduïm l'estat amb 19 emplenaments (esquerra) i amb 18 (centre i dreta), per bé que els sudokus que mostrem ara són normalitzats (cosa que no s'aprecia, perquè ens limitem a les caselles activades i les restants les representem amb 0, per no distreure): a l'esquerra i centre, allò que apareix a la finestra de l'esquerra amb el peu V. E. NORMALITZADA, on 8,5 (**6**) és l'emplenament que s'havia produït en el primer cas però no en el segon, i 4,7 (**3**) la casella amb què s'activa **POST**, subratllada per localitzar-la més fàcilment; el sudoku que presentem a la dreta és el resultat de normalitzar la transposada de la VERSIÓ ESCOLLIDA que apareix a la finestra de la dreta, on l'emplenament que no hem fet se situaria en posició 3,2 i la casella que activa **POST** és 7,8 (**3**).

0 0 0	0 0 0	**4 5** 0
6 0 0	0 0 **9**	0 **1 2**
0 **1** 0	**3** 0 0	0 0 0
8 0 0	0 0 0	0 0 **3**
0 0 0	0 0 **4**	0 **6** 0
0 0 **1**	0 0 0	0 0 0
0 **7** 0	0 0 **5**	0 0 0
0 0 **9**	0 0 **7**	**5** 0 0
0 0 0	0 0 0	0 **4** 0

orig. normalitzada (19)

0 0 0	0 0 0	**4 5** 0
6 0 0	0 0 **9**	0 **1 2**
0 **1** 0	**3** 0 0	0 0 0
8 0 0	0 0 0	0 0 **3**
0 0 0	0 0 **4**	0 0 0
0 0 **1**	0 0 0	0 0 0
0 **7** 0	0 0 **5**	0 0 0
0 0 **9**	0 0 **7**	**5** 0 0
0 0 0	0 0 0	0 **4** 0

orig. normalitzada (18)

0 0 0	0 0 0	0 0 0
0 0 0	0 0 0	**3** 0 0
0 0 **4**	0 **7 5**	0 0 **9**
8 0 0	0 0 0	0 0 **6**
0 **1** 0	0 **9** 0	0 0 0
0 0 0	0 0 **7**	**1** 0 0
3 0 0	0 0 0	0 0 **2**
0 0 0	**4** 0 0	0 **5 1**
0 0 0	0 **5** 0	0 **4** 0

transp. normalitzada (18)

Tot va començar en veure com una espera ja prou incòmoda de **3'10"** es transformava en una plantada en tota regla de **68'15"**, en passar **COMPAT-PERFILS** de treballar amb 19 caselles emplenades a fer-ho amb només 18. Va ser llavors, en adonar-se l'autor que la població dels requadres 9×3 era d'allò més irregular (6 caselles plenes en el primer, 4 en el segon i 8 en el tercer) però que era més regular considerant-la desglossada en requadres 3×9 (6 caselles plenes en el primer, 5 en el segon i 7 en el tercer). Així que va repetir l'operació, però ara aplicant la transformació TR (transposició de files i columnes) al SUDOKU-SOLUCIÓ, i l'espera es va reduir a la cinquena part: **12'40"**. En principi no es va fer el mateix amb l'emplenament de 19 caselles, perquè el grau d'uniformitat o d'homogeneitat (encertadament o no, en la present exposició s'utilitzen ambdós termes indiscriminadament) era exactament la mateixa, desglossant l'ocupació per requadres 9×3 o per requadres 3×9: 6 + 5 + 8 (més endavant es va fer la prova i, malgrat l'aparent irrellevància del canvi, els resultats amb el sudoku transposat van millorar força els de l'original, amb **2'10"** contra els ja esmentats **3'10"**). Però sí que es va provar la transposició en altres dos SUDOKUs-PROBLEMA: el de 21 caselles (emplenat de forma que **POST** s'activés amb 19 emplenaments) i el de 17, utilitzats fa poc per experimentar les excel·lències de l'optimització precedent. Els resultats van ser ben espectaculars: en el de 21, passant d'una distribució per requadres 7 + 4 + 8 a una altra 7 + 7 + 5, passàvem de **6'05"** a **0'15"**; en el de 17, passant d'una distribució per requadres 3 + 7 + 7 a una altra 6 + 6 + 5, passàvem d'haver de desistir (perquè l'ordinador es penjava) a **5'25"**. La relació de causa a efecte no podia ser més clara, per a l'autor: donat un determinat nombre d'emplenaments, com més úniformement distribuït estigui entre els tres requadres 9×3 (o sigui, com més semblant el nombre de caselles emplenades en cada requadre) més semblant serà en tots ells el nombre de tríades de línies compatibles amb els seus emplenaments i, entre aquestes, el de tríades compatibles entre si; així que el nombre de combinacions ternàries que es puguin formar entre els tres conjunts de tríades serà més baix que en circumstàncies diferents i, en conseqüència, també serà menor el nombre de les combinacions compatibles entre si (que, per la primera de les compatibilitats enunciades, comportarà necessàriament compatibilitat amb el conjunt d'emplenaments). Aquesta era, ni més ni menys, la raó que segons l'autor feia recomanable que els emplenaments es distribuïssin de la manera més uniforme possible al llarg i ample de l'escaquer 9×9: que d'aquesta manera, en ser similar el nombre d'emplenaments en els tres requadres 9×3 en el moment de posar-se en marxa **COMPAT-PERFILS**, també ho seria el nombre de tríades de línies compatibles amb aquests emplenaments i, en conseqüència, el nombre de combinacions ternàries que es poguessin formar amb aquestes tríades seria mínim. Us pot resultar difícil de creure però, probablement obnubilat pels èxits assolits en aquesta línia, a l'autor fins i tot li semblava intuitiu que, donat un nombre **k** d'elements, entre les diferents maneres d'agrupar-los en tres conjunts, la mínima quantitat de combinacions ternàries aplegant un element de cada conjunt (qüestió que algebraicament equival a preguntar-se a quins tres valors enters positius de suma constant **k** correspon el menor producte) corresponqués a la formada per tres conjunts de **k**/**3** elements cadascun (o a la distribució més pròxima, si **k** no era divisible per **3**). No cal dir que, a l'hora de demostrar aquest enunciat, cercar-ne alguna referència, trobar-se que l'enunciat correcte era precisament el contrari (que el producte és màxim quan tots tres factors fan **k**/**3**) i quedar-se naturalment amb un pam de nas, no se'n sabia avenir que no se li hagués ocorregut alguna cosa tan elemental com comprovar-ho en un exemple senzill: que **2 + 3 + 4 = 9 = 3 + 3 + 3** i **2 × 3 × 4 = 24 < 27 = 3 × 3 × 3**. Ara, que li tocava demostrar això, encara es va entestar a demostrar una proposició que no era ben bé la mateixa (una prova d'això és que era falsa): que, quan més dispersos fossin els tres valors (en el sentit de màxima diferència entre el més gran i el més petit), menor en seria el producte. No se'n va sortir: referit a dos valors positius **a + b = k** sí que era veritat i es

podia demostrar algebraicament i geomètricament (a partir d'un quadrat de costat **a** i superfície **a × a**, qualsevol rectangle de costats **a > b** té una superfície **a × b** menor i tant més petita quan més gran sigui la diferència **a − b**), però no hi havia manera d'estendre la demostració a tres valors **a + b + c = k** fins que va caure del ruc i es va centrar a demostrar que **a × b × c** era màxim quan **a = b = c** per la via ràpida, sense floritures ni generalitzacions: en el seu *Anàlisis Matemàtico I* de primer de carrera (J. Rey Pastor, P. Pi Calleja i C.A. Trejo; Kapelusz, 7ª edició) havia trobat l'enunciat general, aplicable a qualsevol nombre de reals positius, però la demostració per inducció era equívocament incompleta (pàgina 76), així que va donar per bona la seva pedestre demostració, limitada als tres valors del cas que ens ocupa. Com que ja hem fet prou marrada, la trobareu al final del capítol.

Ara ve el pitjor: si realment jugàvem a treballar amb la pseudomatriu 9×9 ****V*V**** original o amb la seva transposada, de manera que **COMPAT-PERFILS** es trobés sempre amb el màxim nombre de combinacions de tres tríades (referides a cadascun dels requadres 9×3) a processar, ¿com s'explica que l'execució d'aquesta funció acabés sent més curta? Doncs perquè l'efecte d'aquesta acció requeia en primera instància sobre **PERFIL-V*V** i, només secundàriament (en sentit d'ordenació causal i no pas de menysteniment de l'impacte) sobre **COMPAT-PERFILS**. Pel que fa a aquesta, i malgrat l'elevació que localment pugui produir-se en el nombre de files de **L1** compatibles amb una determinada fila de la solució escollida (en quedar sense cap emplenament alguna fila que, en la pseudomatriu original, era una columna buida que diluïa la seva presència entre les 9 files), l'acció provoca una forta reducció en el nombre de combinacions que es poden formar amb els elements dels tres conjunts de línies candidates (que afecta sobretot al requadre 9×3 menys poblat en l'original i també globalment a tot l'escaquer 9×9), la verificació de la compatibilitat interna de les quals és la primera tasca de **COMPAT-PERFILS**. De manera que, quan passem a la següent, que consisteix a verificar la compatibilitat interna de les combinacions que es poden formar amb els elements dels tres conjunts de requadres 9×3 que han passat la selecció, ja ens trobem amb un nombre total d'elements força disminuït: si aquest conjunt es distribuís més o menys equitativament entre els requadres 9×3 inferior, mitjà i superior, llavors sí que seria aplicable l'enunciat algebraic que ens ha fet anar de bòlit i en aquesta fase el procés es faria més llarg perque caldria explorar moltes combinacions; però si, com passa a l'exemple de referència (el de LA VANGUARDIA i **POST** activant-se amb 18 emplenaments), la distribució es fa més desequilibrada, llavors encara hi ha una reducció addicional en aquesta etapa. I, pel que fa a la influència directa sobre l'acció de **PERFIL-V*V**, encara és més fàcil d'entendre: l'ocupació d'una casella condiciona més intensament el perfil de les files del propi requadre 9×3 que no pas el perfil de les altres 6 files (com justificarem de seguida) i, sent així, el condicionament màxim sobre el conjunt de les 9 línies, partint d'un total de **k** emplenaments, correspondrà a un repartiment el més equitatiu possible entre els 3 requadres 9×3 i no a la concentració de la majoria d'ocupacions en un requadre a costa dels altres dos; n'hi ha prou a pensar en el cas extrem de 27 ocupacions concentrades en un mateix requadre, que mai no podran constituir un SUDOKU-PROBLEMA, davant dels molts SUDOKUs-PROBLEMA coneguts amb els 27 emplenaments més o menys regularment distribuïts entre els 3 requadres. Per què diem que l'ocupació d'una casella condiciona més intensament el perfil de les files del propi requadre 9×3 que no pas el de les altres 6 files? Doncs perquè
- en el propi requadre 9×3 condiciona (en el sentit de no poder repetir-se aquest valor) la línia i el requadre 3×3 on se situa (en total, 15 caselles afectades),
- en cadascun dels altres dos requadres només afecta les 3 caselles de la columna.

Tot això pot semblar abstrús i discutible, però es veu més clar referint-ho a un cas concret i amb dades numèriques concretes. Sap greu que, un cop més, hagueu de tornar un o dos fulls enrera per consultar la representació gràfica, però aquí es fa inevitable (tret que anéssim repetint aquesta representació a l'inici de cada full, al llarg de tres pàgines).

Abans de reproduir els llistats que van treient per pantalla **PERFIL-V*V** i **COMPAT-PERFILS** en finalitzar la seqüència de 18 emplenaments, corresponents a la versió original normalitzada (V. E. NORMALITZADA) i a la versió transposada normalitzada (en realitat, la resultant de normalitzar la transposada de VERSIÓ ESCOLLIDA, com havíem dit), repetirem aquí el de la seqüència de 19 (el corresponent a l'annex **Actualització a l'última versió de FES-SOLUCIO del capítol *6- Etapa prèvia: crear una solució, IV (... o a Camprodon)*** del capítol *Un nou camí i també una drecera*), per comparar-lo amb el primer d'ells i apreciar en detall els canvis provocats pel fet d'haver activat **POST** amb una casella de menys:

19 emplenaments (ORIGINAL):

00:00 *Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 196.*
Entre 362880 línies, de compatibles amb la 2ª de la V. E. NORM. n'hi ha 76.
Entre 362880 línies, de compatibles amb la 3ª de la V. E. NORM. n'hi ha 324.
Entre 362880 línies, de compatibles amb la 4ª de la V. E. NORM. n'hi ha 628.
Entre 362880 línies, de compatibles amb la 5ª de la V. E. NORM. n'hi ha 340.
Entre 362880 línies, de compatibles amb la 6ª de la V. E. NORM. n'hi ha 168.
Entre 362880 línies, de compatibles amb la 7ª de la V. E. NORM. n'hi ha 92.
Entre 362880 línies, de compatibles amb la 8ª de la V. E. NORM. n'hi ha 40.
00:05 *Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 96.*

00:15 *Entre 196 línies compatibles amb la 1ª, 76 amb la 2ª i 324 amb la 3ª, hi ha 10232 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

00:35 *Entre 628 línies compatibles amb la 4ª, 340 amb la 5ª i 168 amb la 6ª, hi ha 1920 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

00:35 *Entre 92 línies compatibles amb la 7ª, 40 amb la 8ª i 96 amb la 9ª, hi ha 2496 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

03:10 *Entre les 10232 tríades 1-2-3, les 1920 tríades 4-5-6 i les 2496 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 1117.*

18 emplenaments (ORIGINAL):

00:00 *Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 196.*
Entre 362880 línies, de compatibles amb la 2ª de la V. E. NORM. n'hi ha 100.
Entre 362880 línies, de compatibles amb la 3ª de la V. E. NORM. n'hi ha 400.
Entre 362880 línies, de compatibles amb la 4ª de la V. E. NORM. n'hi ha 1152
Entre 362880 línies, de compatibles amb la 5ª de la V. E. NORM. n'hi ha 1987
Entre 362880 línies, de compatibles amb la 6ª de la V. E. NORM. n'hi ha 256.
Entre 362880 línies, de compatibles amb la 7ª de la V. E. NORM. n'hi ha 120.
Entre 362880 línies, de compatibles amb la 8ª de la V. E. NORM. n'hi ha 40.
00:05 *Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 96.*

00:25 *Entre 196 línies compatibles amb la 1ª, 100 amb la 2ª i 400 amb la 3ª, hi ha 14592 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

32:45 *Entre 1152 línies compatibles amb la 4ª, 1987 amb la 5ª i 256 amb la 6ª, hi ha 15360 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

32:45 *Entre 120 línies compatibles amb la 7ª, 40 amb la 8ª i 96 amb la 9ª, hi ha 3168 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

68:15 *Entre les 14592 tríades 1-2-3, les 115360 tríades 4-5-6 i les 3168 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 21478.*

Observeu que, pel que fa a l'acció de **PERFIL-V*V**, l'augment d'elements de **L1** que són compatibles amb cada línia de ****V*V**** es concentra en el requadre 9×3 mitjà i, en concret, a la línia 5ª, en què la casella 8,5 (**6**) ha deixat d'estar emplenada. La repercusió d'això en el nombre de tríades de línies compatibles entre si (i per tant amb la V. E. NORMALITZADA) també afecta particularment al requadre mitjà (que les multiplica per 8), per bé que la contribució més gran a l'augment de la durada del procés **PERFIL-V*V** + **COMPAT-PERFILS** (que es multiplica per 21) correspon a la verificació de la compatibilitat entre els elements dels tres conjunts de tríades.

Amb la intenció de posar de manifest com les variacions del temps de processat són bàsicament una conseqüència de l'acció de **PERFIL-V*V**, reflectida en les 9 primeres línies d'aquests llistats, segons que hem argumentat abans, tot i que l'obtenció d'aquestes línies sigui pràcticament instantània, farem uns números molt barroers però que resultaran il·lustratius. Si en tots dos llistats realitzem el producte de les xifres mostrades en la 1ª, la 2ª i la 3ª línia, el de les mostrades en la 4ª, la 5ª i la 6ª, i el de les mostrades en la 7ª, la 8º i la 9ª, que és el nombre de combinacions que es poden formar amb les línies de la llista **L1** compatibles amb

la primera, les compatibles amb la segona i les compatibles amb la tercera de cada requadre 9×3 de ****V*V****, i que cal considerar per verificar la seva compatibilitat interna, aquest producte serà una mesura aproximada de la feina que comportarà la verificació (diem aproximada perquè la d'algunes serà abandonada de bon principi, en no encaixar en el patró de SOLUCIÓ NORMALITZADA, mentre que la d'altres seguirà fins al final), vàlida sobretot a efectes de comparació entre un i altre llistat:

- 19 emplenaments (ORIGINAL):
  - Requadre 9×3 inferior: 196 × 76 × 324 = **4.826.304** -> 10.232 (0,212%)
  - Requadre 9×3 mitjà: 628 × 340 × 168 = **35.871.360** -> 1.920 (0,005%)
  - Requadre 9×3 superior: 92 × 40 × 96 = **353.280** -> 2.496 (0,707%)

  Total 41.050.944
- 18 emplenaments (ORIGINAL):
  - Requadre 9×3 inferior: 196 × 100 × 400 = **7.840.000** -> 14.592 (0,186%)
  - Requadre 9×3 mitjà: 1.152 × 1.987 × 256 = **585.990.144** -> 15.360 (0,003%)
  - Requadre 9×3 superior: 120 × 40 × 96 = **460.800** -> 3.168 (0,688%)

  Total 594.290.944

Encara que la diferència entre aquests dos sudokus no és cap transposició sinó una casella pertanyent al requadre 9×3 mitjà i, en conseqüència, els efectes es poden apreciar requadre a requadre, hem realitzat les sumes dels tres productes perquè, en mantenir una significació global, serà l'única informació significativa de què podrem disposar en casos com el que considerarem tot seguit (centrats ja en els 18 emplenaments, comparar l'acció de **PERFIL-V*V** i **COMPAT-PERFILS** sobre l'original i el transposat). Abans, però, aprofitarem per aclarir que la informació addicional que figura a la dreta no és sinó el nombre de combinacions de tres línies que han resultat ser compatibles entre si, i la que teniu entre parèntesis és el tant per cent que representen aquestes tríades respecte al total de combinacions estimades. Com veieu, aquests percentatges són molt irregulars (no és que siguin aleatoris, sinó que depenen de la problemàtica més complexa de la ubicació i contingut de les caselles ocupades), de manera que la correlació entre la informació aportada per **PERFIL-V*V** i el nombre de combinacions de tres tríades (corresponents al requadre 9×3 inferior, mitjà i superior) a considerar, que serà el producte de les xifres de la dreta,

- 19 emplenaments (ORIGINAL): 10.232 × 1.920 × 2.496 = **49.035.018.240** -> -> 1.117 (0,000002%)
- 18 emplenaments (ORIGINAL): 14.592 × 15.360 × 3.168 = **710.053.724.160** -> -> 21.478 (0,000003%)

serà més feble i el percentatge de solucions normalitzades compatibles molt petit.

Passem ara a considerar l'estalvi de temps entre l'aplicació del procés a ****V*V**** i a la pseudomatriu resultant de normalitzar la transposada de ****V**V****, canvi que suposa substituir una distribució d'emplenaments per requadres 9×3 de 6 + 4 + 8 a una de 6 + 5 + 7: (7 + 6 + 5)

18 emplenaments (TRANSPOSADA):

00:00 *Entre 362880 línies, de compatibles amb la 1ª de la V. E. NORM. n'hi ha 146.*
Entre 362880 línies, de compatibles amb la 2ª de la V. E. NORM. n'hi ha 86.
Entre 362880 línies, de compatibles amb la 3ª de la V. E. NORM. n'hi ha 38.
Entre 362880 línies, de compatibles amb la 4ª de la V. E. NORM. n'hi ha 163.
Entre 362880 línies, de compatibles amb la 5ª de la V. E. NORM. n'hi ha 378.
Entre 362880 línies, de compatibles amb la 6ª de la V. E. NORM. n'hi ha 256.
Entre 362880 línies, de compatibles amb la 7ª de la V. E. NORM. n'hi ha 6.
Entre 362880 línies, de compatibles amb la 8ª de la V. E. NORM. n'hi ha 983.
00:05 *Entre 362880 línies, de compatibles amb la 9ª de la V. E. NORM. n'hi ha 6985*

00:05 *Entre 146 línies compatibles amb la 1ª, 86 amb la 2ª i 38 amb la 3ª, hi ha 864 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

00:20 *Entre 163 línies compatibles amb la 4ª, 378 amb la 5ª i 256 amb la 6ª, hi ha 30408 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

00:45 *Entre 6 línies compatibles amb la 7ª, 983 amb la 8ª i 6985 amb la 9ª, hi ha 3932 tríades normalitzades compatibles amb la V. E. NORMALITZADA.*

12:40 *Entre les 864 tríades 1-2-3, les 30408 tríades 4-5-6 i les 3932 tríades 7-8-9, de solucions normalitzades compatibles amb la V. E. NORMALITZADA n'hi ha 10277.*

Per no fer això encara més llarg, no hem repetit el llistat relatiu a la ****V*V**** original abans del corresponent a la resultant de normalitzar la transposada de ****V**V**** (això és el que farem més endavant, quan implementem la substitució el el programa, però el que hem fet ara és aplicar a la SOLUCIÓ COPIADA la transformació TR, per obtenir VARIANT ESCOLLIDA i V. E. NORMALITZADA), però sí que repetirem els números que abans hem titllat de barroers però que aporten un plus d'informació:

- 18 emplenaments (ORIGINAL):
  - Requadre 9×3 inferior: 196 × 100 × 400 = 7.840.000 -> 14.592 (0,186%)
  - Requadre 9×3 mitjà: 1.152 × 1.987 × 256 = 585.990.144 -> 15.360 (0,003%)
  - Requadre 9×3 superior: 120 × 40 × 96 = 460.800 -> 3.168 (0,688%)

  Total **594.290.944**
- 18 emplenaments (TRANSPOSADA):
  - Requadre 9×3 inferior: 146 × 86 × 38 = 477.128 -> 864 (0,181%)
  - Requadre 9×3 mitjà: 163 × 378 × 256 = 15.773.184 -> 30.408 (0,193%)
  - Requadre 9×3 superior: 6 × 983 × 6.985 = 41.197.530 -> 3.932 (0,010%)

  Total **57.447.842**

Ara no hem remarcat en negreta el nombre de combinacions a considerar en cadascun dels requadres 9×3, sinó el total, perquè es tractava de dues ****V*V**** d'estructura ben dispar. La sobtada pujada en el nombre de combinacions del requadre superior es deu al fort augment del nombre de files de la llista **L1** compatibles amb la 9ª, que és completament buida (l'única de tots tres casos, com podeu comprovar en les representacions de cinc fulls enrera), però malgrat això el nombre de combinacions de primeres, segones i terceres línies d'aquest requadre és 14 vegades menor que en el requadre mitjà del sudoku sense transposar. Això té sentit perquè, tot i ser aquests requadres els més despoblats, el primer té 5 caselles ocupades i el segon només 4, però el fet que la 8ª línia de l'original tingui a **L1** més candidates que la 7ª de la transposada, malgrat tenir les dues 4 emplenaments, és desconcertant, perquè a sobre el primer pertany a un requadre 9×3 amb 8 caselles plenes mentre que el del segon només en té 5: caldrà baixar al detall dels valors admissibles en primera instància a cadascuna de les caselles lliures d'aquestes línies, i tenir present que les 5 caselles buides de la 8ª línia del sudoku original admeten un total de **3 × 3 × 4 × 4 × 3 = 432** combinacions de valors (perquè dels 5 valors encara absents n'hem d'excloure 2 a la 1ª casella lliure, 2 a la 2ª, 1 a la 3ª, 1 a la 4ª i 2 a la 5ª) mentre que les de la 7ª línia del transposat n'admeten un total de **3 × 4 × 4 × 3 × 2 = 288** (perquè dels 5 valors encara absents n'hem d'excloure 2 a la 1ª casella lliure, 1 a la 2ª, 1 a la 3ª, 2 a la 4ª i 3 a la 5ª), per recuperar la lògica d'aquest trencaclosques.

Quant al nombre de combinacions de tríades, no ha davallat en la mateixa proporció que l'esmentada reunió dels tres conjunts de combinacions però, Déu n'hi do!: s'ha reduït a la setena part (si parlem de temps, a la cinquena). Heus aquí els valors:
- 18 emplenaments (ORIGINAL): 14.592 × 15.360 × 3.168 = **710.053.724.160** -> -> 21.478 (0,000003%)
- 18 emplenaments (TRANSPOSADA): 864 × 30.408 × 3.932 = **103.303.517.184** -> -> 10.277 (0,000010%)

A l'hora de prendre la decisió d'incorporar al mètode 2 l'artifici de transposició automàtica, es presentava el dubte de si fer-ho només en treure **FOTO1** (intervenció de **PERFIL-V*V** i **COMPAT-PERFILS**) o també en treure **FOTO2** (intervenció d'**ANORMALS**), i ja hem avançat cinc fulls enrera, en presentar aquesta possibilitat de millora, que finalment havíem desestimat de fer-ho. Ara confessarem que la intenció inicial va ser una altra perquè, malgrat que en el segon cas la relació de causa a efecte no fos tan clara com en el primer, els primers resultats experimentals eren prou satisfactoris. Però només era qüestió de seguir insistint perquè, sense necessitat de sortir de l'àmbit dels exemples ja esmentats, l'aparició d'un contraexemple va aconsellar fer marxa enrera. Veiem-lo, esguerrant tota una sèrie de proves prèvies que semblaven indicar que també, quan més uniformement estiguessin distribuïts els emplenaments per requadres 9×3, més ràpidament seria processada **ANORMALS** (sempre que es tornés a avaluar la distribució en el moment previ a l'execució la funció, cosa que no podia confiar-se a l'aplicació manual de la transformació TR, i que ha requerit la construcció d'una versió *ad hoc* que finalment ha estat desestimada):
- En el sudoku de 17 (el primer de la col·lecció publicada per Gordon Royle) hem dit que sense transposició l'ordinador es penjava a **COMPAT-PERFILS**, així que no ha estat possible arribar a **ANORMALS**. Amb transposició automàtica, sí que hi hem pogut arribar (com que els dos processos es posen en marxa sobre el mateix estat

d'emplenament, un després de l'altre, aquest és l'únic dels exemples en què era indiferent efectuar la transposició manualment o confiar-la a l'automatitzada), amb el resultat d'execució d'**ANORMALS** pràcticament instantània.

- En l'exemple que hem il·lustrat més detalladament (el de LA VANGUARDIA), tampoc quedava clara la qüestió, perquè s'hi barrejaven massa coses diferents. Passant per l'emplenament de la casella 8,5 (**6**), de forma que l'activació de 4,7 (**3**) fos la 19ª, ja hem dit que no es produïa cap transposició automàtica, perquè el grau d'uniformitat era el mateix si consideràvem l'ocupació per requadres 9×3 o per requadres 3×9: 6 + 5 + 8 i 6 + 5 + 8. Si malgrat això forçàvem la transposició usant la transformació TR, l'execució de **PERFIL-V*V** + **COMPAT-PERFILS** passava de **3'10"** a **2'10"**, i la d'**ANORMALS** passava de **0'20"** a ser pràcticament instantània, si bé cal precisar que, treballant amb el sudoku original normalitzat, la funció s'executava en emplenar la penúltima casella del SUDOKU-PROBLEMA (22ª: 4,2 (**1**)) mentre que, en fer-ho amb el transposat normalitzat, s'executava en emplenar-ne l'última (23ª: 5,2 (**4**) o 5,9 (**4**), segons que la localitzem en l'original o en el trasposat, ambdós normalitzats). De tota manera, això no és sinó una especulació que no compta com a contraexemple perquè, fos quina fos la casella que desfermés l'execució d'**ANORMALS**, la distribució dels emplenaments seria tan homogénia en el sudoku original com en el transposat (amb la 22ª, la distribució a l'original seria 8 + 6 + 8 i en el transposat 8 + 6 + 8; amb la 23ª, la distribució en el transposat seria 8 + 6 + 9 i a l'original 9 + 6 + 8), així que tampoc s'escauria la transposició del sudoku.
- Si ara utilitzem el mateix sudoku per emplenar-lo saltant-nos la casella que en versió original normalitzada és 8,5 (**6**) i en versió transposada normalitzada és 3,2 (**6**), com mostràvem en les representacions centre i dreta de la pàgina 544, ja hem explicat que passava en arribar al divuitè emplenament: en veure que la transposició donava lloc a una distribució més homogènia (6 + 5 + 7 enfront de 6 + 4 + 8), l'acció de **PERFIL-V*V** i **COMPAT-PERFILS** s'aplicava a la transposada normalitzada (7 + 6 + 5), obtenint una reducció de temps considerable (de **68'15"** a **12'40"**). Els llistats que ens proporcionaven aquestes funcions també s'havien reproduït, culminant en el nombre de solucions normalitzades compatibles amb la V. E. NORMALITZADA (un petit detall és que aquesta referència, del tot impecable quan l'usuari decideix lliurement que la VERSIÓ ESCOLLIDA sigui la transposada de la SOLUCIÓ CREADA/COPIADA/CANÒNICA, adoptant transformació TR per obtenir-la -que és el que hem fet nosaltres abans-, hauria de canviar quan la substitució de la V. E. NORMALITZADA pel resultat de normalitzar la transposada de la VERSIÓ ESCOLLIDA fos un procés automàtic, del qual convé informar l'usuari, si més no; però aquesta és una altra història, que abordarem així que acabem aquest incís), que en un cas era de **21.478** i en l'altre de **10.277**, però cal tenir present que aquí no s'acabava el procés: per arribar a determinar el SUDOKU-PROBLEMA mostrat a LA VANGUARDIA, encara queden 5 caselles per emplenar, però mentre treballant amb el sudoku original **ANORMALS** intervé quan emplenem la penúltima, fent-ho amb el transposat aquesta funció no intervé fins que no arribem a l'última casella. Veiem en ambdós casos els 6 últims emplenaments (expressats en les coordenades respectives), el nombre de solucions compatibles normalitzades (N) i totals (T) després de cada un, i la intervenció de **PERFIL-V*V** + **COMPAT-PERFILS** i **ANORMALS**:

sobre v. orig. normalitzada	sobre v. transp. normalitzada
4,7 (**3**) ->	4,7 (**3**) ->
(**PERFIL-V*V** + **COMPAT-PERFILS**)	(**PERFIL-V*V** + **COMPAT-PERFILS**)
4,7 (3) -> **21.478** N	6,8 (3) -> **10.277** N
8,5 (**6**) -> **1.117** N	3,2 (**6**) -> **1.542** N
4,6 (**7**) -> **325** N	1,8 (**7**) -> **1.542** N
4,2 (**1**) -> **13** N	5,8 (**1**) -> **124** N
4,1 (**6**) -> **1** N ->	4,8 (**6**) -> **2** N
(**ANORMALS**)	5,9 (**4**) -> **1** N ->
4,1 (6) -> **593** T	(**ANORMALS**)
5,2 (**4**) -> **1** T	5,2 (4) -> **1** T

Cal aclarir que les dades de l'esquerra corresponen a la versió anterior, que en anar a executar **PERFIL-V*V** + **COMPAT-PERFILS** no fa cap estudi comparatiu de la distribució d'emplenaments a v. orig. normalitzada i a v. transp. normalitzada, raó per la qual tot s'esdevé en l'àmbit de la primera. Per contra, les dades de la dreta corresponen a la versió *ad hoc* en què aquesta avaluació s'efectua en anar a executar **PERFIL-V*V** + **COMPAT-PERFILS** i en anar a executar **ANORMALS**: la primera dóna com a resultat que la distribució 6 + 4 + 8 (orig. norm.) és pitjor que 6 + 5 + 7 (la seva transposada), s'efectua la transposició i normalització de l'original (7 + 6 + 5) i se segueix treballant amb aquesta, amb l'estalvi que hem esmentat; la segona dóna com a resultat que la distribució 9 + 6 + 8 (orig. norm.) té el mateix ordre d'uniformitat que 6 + 9 + 8 (la seva transposada) i es

torna a treballar amb la primera. Pel que fa al temps d'espera, corresponent a **PERFIL-V*V** + **COMPAT-PERFILS** i a **ANORMALS**, en el procés representat a l'esquerra és de **68'15"** + **0'20"** = **68'35"** i en el de la dreta és de **12'40"** + **0'0"** = **12'40"**. Ara veiem, comparant aquests resultats amb els del cas precedent (primera parada als 19 emplenaments, en lloc dels 18 d'ara), en què la durada d'**ANORMALS** també era de **0'0"** quan l'ús de la transformació TR garantia que aquesta funció actués sobre una versió transposada del sudoku (que sigui la transposada de l'original normalitzada o el producte de normalitzar la transposada de l'original tampoc no és rellevant), que la dràstica reducció d'aquesta durada no té res a veure amb el dilema transposició/no-transposició, sinó en el fet d'haver de bregar amb 593 solucions compatibles o amb una de sola (només per constatar això hem castigat al sofert lector descrivint amb un cert detall aquesta variant), raó per la qual aquest cas tampoc no és concloent com a exemple ni com a contraexemple.

- Haurem d'anar al quart del exemples esmentats per trobar un contraexemple sense fissures, que ens permetrà veure com la intervenció d'**ANORMALS** es fa més llarga actuant sobre la versió transposada que actuant sobre l'original (com que usarem la versió provisional *ad hoc*, que també aplica el criteri de major uniformitat per requadres 9×3 sobre aquesta funció, les dues versions seran normalitzades, tot i que aquí es dóna la circumstància que original i transposada ja en són, de normalitzades), malgrat que es produeix en una mateixa situació d'emplenament. Ens referim al de 21 emplenaments, del qual ja havíem dit que la transposició a **PERFIL-V*V** + **COMPAT-PERFILS**, efectuada per passar d'una distribució de requadres 7 + 4 + 8 a una altra 7 + 7 + 5, ens permetia reduir de **6'05"** a **0'15"** la durada de l'execució d'aquest tàndem, i perquè no hi hagi cap mena de dubte detallarem els tres processos que es comparen. Aclarint que, igual que abans, els epígrafs subratllats es refereixen a la versió de la pseudomatriu 9×9 amb què treballaran **PERFIL-V*V**, **COMPAT-PERFILS** i l'actualització **NUMCOMPAT** en cada emplenament, fins arribar a **ANORMALS** (ras i curt, el procés comprès entre **FOTO1** i **FOTO2**), teniu a l'esquerra la utilització d'una versió del programa anterior a la problemàtica que ara ens ocupa (sempre juga amb el sudoku original normalitzat), en el centre la versió que únicament avalua si la distribució d'emplenaments per requadres 3×9 és més uniforme que la de requadres 9×3 en anar a **PERFIL-V*V** (transposant en cas afirmatiu la pseudomatriu de treball i restituint l'original en arribar a **ANORMALS**) i a la dreta la versió *ad hoc* que torna a avaluar la distribució en arribar a **ANORMALS**, segona avaluació que en l'exemple actual conduirà a seguir treballant amb la transposada, perquè la distribució per requadres és 7 + 6 + 8 en la pseudomatriu original i 7 + 7 + 7 en aquesta última. La columna central i la de la dreta (en totes dues es reconeix la transposició perquè les coordenades de les caselles emplenades entre **FOTO1** i **FOTO2** estan permutades en relació a les de l'esquerra) es diferencien només en les coordenades de l'última línia, que a la central tornen al sistema original i a la dreta es mantenen en el transposat:

sobre v. orig. normal.	sobre v. transp. normal.	sobre v. transp. normal.
7,5 (**1**) ->	7,5 (**1**) ->	7,5 (**1**) ->
(**PERFIL-V*V**+**COMPAT-PERF.**)	(**PERFIL-V*V**+**COMPAT-PERF.**)	(**PERFIL-V*V**+**COMPAT-PERF.**)
7,5 (1) -> **218** N	5,7 (1) -> **121** N	5,7 (1) -> **121** N
8,5 (**2**) -> **24** N	5,8 (**2**) -> **12** N	5,8 (**2**) -> **12** N
9,5 (**7**) -> **1** N ->	5,9 (**7**) -> **1** N ->	5,9 (**7**) -> **1** N ->
(**ANORMALS**)	(**ANORMALS**)	(**ANORMALS**)
9,5 (7) -> **1** T	9,5 (7) -> **1** T	5,9 (7) -> **1** T

Malgrat l'aparent semblança, les esperes corresponents als dos dispositius que sovint hem qualificat de maquinària pesant són: **6'05"** + **0'00"** el el procés que representem a l'esquerra, **0'15"** + **0'00"** en el del centre i **0'15"** + **0'15"** en el de la dreta, resultats que deixen palès que en el present exemple, un cop més, la substitució de ****V*V**** per la transposada normalitzada de ****V**V**** (en les circumstàncies de tots conegudes i que no repetirem) permet de reduir la durada de **COMPAT-PERFILS**, però el mateix tractament no té cap eficàcia sobre **ANORMALS**. I aquí ja no s'hi val a atribuir el fracàs a factors aliens als elements en joc.

A l'hora de posar fil a l'agulla, hem de subratllar que la transposició ha de ser transparent per a l'usuari, a qui ja haurem marejat prou explicant-li el perquè de l'ús d'una V. E. NORMALITZADA de la VERSIÓ ESCOLLIDA. Però una ocultació absoluta tampoc no és recomanable, perquè es podria donar el cas d'algú una mica observador a qui causés perplexitat veure com un mateix estat d'emplenament es tradueix en un nombre diferent de solucions normalitzades compatibles amb la V. E. NORMALITADA, si l'ordre d'activació de caselles no havia estat ben bé igual. No cal anar massa lluny: en l'exemple de LA VANGUARDIA hem vist com, un cop emplenades les mateixes 19 caselles, podíem tenir 1.117 solucions normalitzades compatibles (si la casella 8,5 (**6**) l'havíem activat abans que la 4,7 (**3**)) o 1.542 (si ho havíem fet després).

Així que, quan s'esdevingui la transposició automàtica, alguna explicació hauríem de donar, si més no. Una manera acceptable de nedar i guardar la roba és incloure algun aclariment del tipus

****
Per tal d'abreujar aquest procés, s'ha substituït la VERSIÓ ESCOLLIDA per la seva transposada (substitució de files per columnes), però com que seria poc entenedor al·ludir a "solucions normalitzades compatibles amb el resultat de normalitzar la transposició de la VERSIÓ ESCOLLIDA", no s'ha canviat el text.
****
A partir d'ara, l'expressió "solucions normalitzades" s'haurà d'entendre així.
****

quan, en haver acabat la seva tasca, **COMPAT-PERFILS** presenti els últims resultats i commuti a pantalla de text, just abans de **(SEGUIR ())** (naturalment, sols si s'ha produït la transposició de ****V**V****). Més pelut serà haver de bregar amb una altra versió normalitzada, que l'usuari no necessitarà visualitzar però que el programa sí que necessita, i que anomenarem **T**V*V**** (en realitat amb 3, comptant ****9*9****).

Com ja hem dit, no hem de transposar la V. E. NORMALITZADA (això, en general, no donaria cap solució normalitzada), sino de normalitzar la transposada de la VERSIÓ ESCOLLIDA, raó per la qual **T**V*V**** no serà normalment la transposada de ****V*V****. A més, cal no oblidar que, si a la dreta hem de veure en gris ****9**9**** i en blanc ****V**V****, i a l'esquerra hem de veure en gris ****9*9**** i en blanc ****V*V**** (no només hem de conèixer el valor assignat a cada casella sinó visualitzar-lo en les dues finestres i fer-ho amb una lluminositat que ens indiqui si aquest valor és públic o no), també és imprescindible jugar alhora amb aquestes quatre pseudomatrius 9×9, mantenint-les actualitzades de forma permanent. Per sort no passa el mateix amb la transposada normalitzada **T**V*V****, que en determinades circumstàncies (**TRANSP** = **T**) substituirà ****V*V**** en **PERFIL-V*V** + **COMPAT-PERFILS** però que serà transparent per a l'usuari: arrossegarà una existència virtual i solitària, sense la companyia gris i permanent d'una **T**9*9**** a l'esquerra i sense necessitat de crear cap **T**9**9**** ni **T**V**V**** (serien les transposades de ****9**9**** i ****V**V****) a la dreta. Només al començament convindrà definir **T**9*9**** per poder trobar **TN1**, **TN2**, **TN3** i **TN4**, però després ja no n'haurem de fer res i per això la declararem com a variable local a **NORMTEXT-9*9**. Inicialitzada ****V*V****, per actualitzar-la amb cada clic amb **COMMUTA** només necessitem conèixer les coordenades **X** i **Y** de la casella activada en aquesta pseudomatriu. Ara bé, com que no coneixem la correspondència directa entre ****V*V**** i **T**V*V****, haurem de partir de la posició de la casella a ****V**V**** (donada per **Q**, si hem fet clic a la finestra dreta, o per **R** si l'hem fet a la finestra esquerra), permutar les cordenades (així tindrem les de la transposada de ****V**V****) i trobar la **Y** que li pertocarà mitjançant la nova funció **TV->TN**. Però ja estàvem avançant esdeveniments i parlant de variables estranyes (**TRANSP**, **TN1**, **TN2**, **TN3** i **TN4**) com si les coneguéssim de tota la vida, així que més valdrà posar-hi fre i justificar sistemàticament les adaptacions del programa, abans de presentar el codi afectat:

- El cens de població (caselles ocupades) i la seva distribució per requadres 9×3 i 3×9 anirà a càrrec de la funció **DET-TRANSP** que, igual com passava a l'inici de **MASCARA**, usará **N-3*3** per saber quantes caselles emplenades té cada requadre 3×9. Com el lector haurà suposat, **TRANSP** és el senyal binari lligat a la propietat de ser més uniforme la distribució per requadres 3×9 (cal aplicar la transposició), i el fet que hi hagi una variable homònima a la fase de presolució del programa (que a sobre és local a **TERCETS** i a **FILS/COLS**) no presenta cap risc de confusió.
- Pel que fa a decidir en quines circumstàncies fem el cens, de les consideracions que hem fet abans es dedueix la conveniència de substituir files per columnes en ****V*V**** únicament en posar-se en marxa la maquinària **PERFIL-V*V** + **COMPAT-PERFILS** (amb el mètode 2, en activar **SUD-MIN** el senyal **POST** i sempre que, abans d'haver tret **FOTO2**, desactivem alguna de les caselles que sortien a **FOTO1**), raó per la qual hem situat l'accés a **DET-TRANSP** a l'inici de la primera funció del tàndem. En la implementació inicial, el dispositiu s'havia construït així, com a funció independent, per poder donar servei a **PERFIL-V*V** i a **ANORMALS**. I ara que podríem transferir-ne el contingut a l'inici de la primera funció, com que ja la tenim prou farcida hem decidit de no tocar-ho.
- A **NORMTEXT-9*9** les funcions **VAR->NORM** i **PRE-NORM->VAR** (modificades) s'executen dos cops, per tal que, a més de la pseudomatriu 9×9 ****9*9**** (versió normalitzada de la variant escollida ****9**9****) i de les variables **N1**, **N2**, **N3** i **N4** (paràmetres que ens permetran obtenir coordenades Y de la 1ª a partir de la 2ª i viceversa, mitjançant **N->V**, **V->N** i les funcions auxiliars **Y1**, **Y2** i **Y3**), obtinguem **T**9*9**** (versió normalitzada de la variant escollida ****9**9**** transposada) i les noves variables **TN1**, **TN2**, **TN3** i **TN4** (paràmetres que ens permetran obtenir coordenades **Y** de la 2ª a partir de la nova funció **TV->TN** i de les mateixes auxiliars).

- A **FES-SUDOKU**, un cop haguem inicialitzat ****V*V**** (fent-la igual a ****9*9****) però abans de començar les iteracions **while**, inicialitzarem també **T**V*V****, amb una aplicació simplificada de **TRANSPOSA-HO** incorporada en el codi d'aquesta funció.
- L'actualització de **T**V*V**** la farem a **CLIC**, després d'haver actualitzat ****V*V**** i ****V**V****, amb independència que el senyal **TRANSP** estigui activat o no. Tant si el clic s'ha produït a la finestra esquerra (possibilitat limitada al mètode 2) com si ho ha fet en el de la dreta, farem una cosa que pot semblar rocambolesca però que a la pràctica no ho és: partir de les coordenades de la segona finestra i, mitjançant la nova funció **TV->TN** abans esmentada, calcular les de la versió normalitzada, que anomenarem **TP**; un cop fet això, **COMMUTA** actualitzarà **T**V*V****.
- El procediment per prendre **TP** en comptes de **P** i **T**V*V**** en comptes de ****V*V****, quan el senyal **TRANSP** estigui activat, és fàcil: a l'àmbit de **FES-SUDOKU-234**, **P** segueix figurant a tot arreu però **VTV** substitueix ****V*V****; així, si **TRANSP** està activat, **P** serà **TP** i **VTV** serà **T**V*V****; si no, **P** serà ell mateix i **VTV**, ****V*V**** (a diferència de **P**, caldrà un nom diferent de **T**V*V**** i de ****V*V****, perquè cal mantenir diferenciades i permanentment actualitzades ambdues pseudomatrius 9×9).

Aquí teniu el codi implicat en aquesta última optimització. En totes les funcions preexistents (que reproduïm totalment o parcial, segons el grau d'afectació) s'han subratllat les parts afegides o modificades, per localitzar-les més ràpidament:

```
(defun N-3*3 (9*9 P-SEMPRE / 3*3 *3 I)
  (setq K -1)
  (repeat 3
    (setq *3 ())
    (repeat 3
      (setq J -1 K (1+ K))
      (repeat 3
        (setq N 0)
        (repeat 3
          (setq J (1+ J) I (nth J (nth K 9*9)))
          (if (or (atom I) (and P-SEMPRE (equal I P))) (setq N (1+ N))))
        (setq *3 (cons N *3))))
    (setq *3 (list (+ (nth 2 *3) (nth 5 *3) (nth 8 *3))
                   (+ (nth 1 *3) (nth 4 *3) (nth 7 *3))
                   (+ (nth 0 *3) (nth 3 *3) (nth 6 *3))) 3*3 (cons *3 3*3)))
  (setq 3*3 (reverse 3*3)))

(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP*)
  (setq N (N-3*3 (if 1-2 **V**V** **9*9**) T) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ))
  ..........................................................)

(defun VAR->NORM (/ K L 9**9 A B C D)
  (setq **9**9** (TRANSPOSAR **9**9**) K -1)
  (repeat 3
    (setq L ())
    (repeat 3 (setq K (1+ K) L (cons (nth K **9**9**) L)))
    (ORDENA-3)
    (foreach E L (if E (setq L (list A E D))))
    (setq 9**9 (append 9**9 (list L))))
  (setq K -1 L ())
  (foreach E 9**9 (setq L (cons (cons (caar E) E) L)))
  (ORDENA-3)
  (foreach E L (if E (setq **9*9** (append (cdr A) (cdr E) (cdr D)))))
  (if (not T**9*9**) (setq T**9*9** **9*9**))

(defun PRE-NORM->VAR (/ L* L** N F F1 F2 F3 F4)
  (foreach L  (reverse **9*9**) (setq L*  (cons (car L) L*)))
  (foreach L (reverse **9**9**) (setq L** (cons (car L) L**)))
  (setq N1 (- 9 (length (member (nth 0 L*) L**))) F1 (/ N1 3)
        N2 (- 9 (length (member (nth 3 L*) L**))) F2 (/ N2 3)
        ....................................................
        F3 (- 9 (length (member (nth 8 L*) L**))) F3 (rem F3 3)
        N3 (list F1 F2 F3) N3 (PERMUT-N N3))
  (if (not TN1) (setq TN1 N1 TN2 N2 TN3 N3 TN4 N4))
```

```
(defun NORMTEXT-9*9 (/ T**9*9**)
  (prompt "\n\n\nSi anomenem SOLUCIÓ NORMALITZADA aquella en què les files de ")
  ....................................
  (prompt " ja queda determinada.")
  (SEGUIR ())
  (repeat 2 (VAR->NORM) (PRE-NORM->VAR))
  (graphscr)
  ....................................
  (prompt "i després seguir com si no res.")
  (SEGUIR ()))

(defun TV->TN (N1 N2 N3 N4 / N)
  (cond ((< Y 3) (Y1 3 2 3 4 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 4 1 3))
        (T (Y3 3 1 4 3 5)))
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 3 4 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 4 1 3))
        (T (Y3 1 1 4 3 5))))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if (and (> 1-2 "1") Q) (setq P Q X (car P) Y (cadr P) PX QX PY QY))
        ..............
        (if (> 1-2 "1")
            (progn
              (command "CVPORT" 2)
              .....................
              (command "CVPORT" 3))))
      (progn
        (setq N (ELEMENT **9*9**) ...)
        ...............................
        (setq P Q X (car P) Y (cadr P) PX QX PY QY Q ())))
  (if (and (= 1-2 "2") (not FOTO2))
      (progn
        (setq TP (reverse (if (> (getvar "CVPORT") 2) Q R))
              X (car TP) Y (cadr TP))
        (TV->TN TN1 TN2 TN3 TN4)
        (setq T**V*V** (COMMUTA T**V*V**) TP (list X Y)
              VTV (if TRANSP T**V*V** **V*V**) P (if TRANSP TP P)))
      (if (> 1-2 "1") (setq VTV **V*V**))))

(defun FOTO-V*V (/ FOTO)
  (setq J -1)
  (foreach L VTV
    (setq I -1 J (1+ J))
    (foreach E L
      (setq I (1+ I))
      (if (atom E) (setq FOTO (cons (list I J) FOTO)))))
  FOTO)

(defun DET-TRANSP (/ N Ñ)
  (setq N (N-3*3 **V*V** ()) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ) ; També es pot fer amb **V**V**.
        TRANSP (> (- (eval (cons 'max N)) (eval (cons 'min N)))
                  (- (eval (cons 'max Ñ)) (eval (cons 'min Ñ))))
        VTV (if TRANSP T**V*V** **V*V**) P (if TRANSP TP P)))

(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V KLINIA-V*V KKE KE)
  (if (not L1) (C:CARREGA-123456789))
  (DET-TRANSP)
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
  (repeat 3
    (setq J (- J 9) K (+ K 3))
    (repeat 3
      (setq J (+ J 3) K (- K 3) W ())
      (repeat 3
        (setq J (- J 3) K (1+ K))
```

```
        (repeat 3
          (setq J (1+ J) V (nth J (nth K VTV)))
          (if (atom V) (setq W (cons V W))))
      (cond ((= J 2) (setq A W))
            ((= J 5) (setq B W))
            (T       (setq C W))))
    (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
  (setq L1C (reverse L1C) J -1)
  (repeat 9
    (setq J (1+ J) K -1 L ())
    (repeat 9 (setq K (1+ K) C (nth J (nth K VTV))
                    L (if (atom C) (cons C L) L)))
    (setq L2C (cons L L2C)))
  (setq L2C (reverse L2C) LINIES-V*V () KLINIES-V*V () K -1)
  (terpri)
  (repeat 9
    (setq LINIA-V*V () KLINIA-V*V () L "" K (1+ K))
    (mapcar '(lambda (E F G)
               (setq L (strcat L (if (or F G)
                                     (if (atom E)
                                         (itoa E)
                                         (progn
                                           (setq C "]")
                                           (foreach H (append F G)
                                             (setq C (strcat (itoa H) C)))
                                           (strcat "[~" C)))
                                     "#"))))
            (nth K VTV) (nth K L1C) L2C)
    ..................................................................)
  (setq LINIES-V*V (reverse LINIES-V*V) KLINIES-V*V (reverse KLINIES-V*V)
        L1 () *KK* (reverse *KK*))
  (terpri))

(defun ACLARIMENT ()
  (terpri) (***)
  (prompt "\n Per tal d'abreujar aquest procés, s'ha substituït la VERSIÓ")
  (prompt " ESCOLLIDA per la\n seva transposada (substitució de files per")
  (prompt " columnes), però com que seria poc\n entenedor al•ludir a \"")
  (prompt "solucions normalitzades compatibles amb el resultat de\n normalitzar")
  (prompt " la transposició de la VERSIÓ ESCOLLIDA\", no s'ha canviat el text.")
  (***)
  (prompt "\nA partir d'ara, l'expressió \"solucions normalitzades\"")
  (prompt " s'haurà d'entendre així.")
  (***) (terpri))

(defun COMPAT-PERFILS (/ L1 LL0 KLL0 LL1 KLL1 L2 LL2 KLL2 KT01 3L M N TERCETS-V*V
                         CC0 PCC0 CC1 PCC01 CC2 CCC1 CCC2 TT0 TT1 TT2 LOCT1 LOCT2
                         TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  ....................................
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq KLINIES-V*V () TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        ....................................................................
        (textscr)
        (if TRANSP (ACLARIMENT))
        (SEGUIR ())
        (graphscr))))

(defun ANORMALS (NORM=1 1-2<34 / L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P PRE
                                 NORM)
  (if TRANSP (setq VTV **V*V** TRANSP ()))
  (setq L0A8 '(0 1 2 3 4 5 6 7 8) FOTO1 () FOTO2 (FOTO-V*V))
  ....................................
  (if (and NORM=1 (= KK 1)) (setq KKK ())))
```

```
(defun FES-SUDOKU-234 (/ LINIES-V*V KLINIES-V*V *K* *KK* R RX RY TP)
  (setq Q ())
  (CLIC)
  (if POST
      (if V
          (if (or (member P FOTO1) (member P FOTO2))
              (if (= 1-2 "4") ...................)
              (progn
                (setq I (assoc P S-V*V) S-V*V (SUPR I S-V*V)
                      I (caar S-V*V) J (nth (car I) (nth (cadr I) VTV)))
                (NUMCOMPAT I J ())))
          (progn .................)))
  (if (not POST) ..................)
  (if POST .......................))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     T**V*V** **V**V** VTV TRANSP)
  (if (= 1-2 "1") ...)
  (setq .............)
  (prompt (strcat "\n.\n.\n" GR))
  (if (= 1-2 "2")
      (progn
        (foreach EE (reverse (TRANSPOSAR **V*V**))
          (setq J ())
          (foreach E EE (setq J (cons (reverse E) J)))
          (setq T**V*V** (cons (reverse J) T**V*V**))))
      (if (= 1-2 "4") (setq FOTO2 VTV **V*V** (FOTO-V*V) POST T)))
  (repeat 2 .......................................)
  ................................................
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" ""))
```

Observeu que el final de la funció **CLIC** l'hauríem pogut deixar així

```
  ..............................
  (if (and (= 1-2 "2") (not FOTO2))
      (progn
        (setq TP (reverse (if (> (getvar "CVPORT") 2) Q R))
              X (car TP) Y (cadr TP))
        (TV->TN TN1 TN2 TN3 TN4)
        (setq T**V*V** (COMMUTA T**V*V**) TP (list X Y)
              VTV (if TRANSP T**V*V** **V*V**) P (if TRANSP TP P)))))
```

actuació amb què ja n'hi hauria hagut prou per evitar que, més enllà de l'execució d'**ANORMALS**, se seguissin repetint inútilment aquestes instruccions. Però, com ja haureu vist, hem optat per deixar-lo d'aquesta altra forma:

```
  ..............................
  (if (and (= 1-2 "2") (not FOTO2))
      (progn
        (setq TP (reverse (if (> (getvar "CVPORT") 2) Q R))
              X (car TP) Y (cadr TP))
        (TV->TN TN1 TN2 TN3 TN4)
        (setq T**V*V** (COMMUTA T**V*V**) TP (list X Y)
              VTV (if TRANSP T**V*V** **V*V**) P (if TRANSP TP P)))
      (if (> 1-2 "1") (setq VTV **V*V**))))
```

El motiu és que la funció **FES-SUDOKU-234** és accedida des dels mètodes 2, 3 i 4, i gairebé tots els trams de codi són compartits per més d'un d'ells, de manera que, si els accessos des de les etapes "PRE-FOTO" i "FOTO 1-2" del primer es nodrissin amb **VTV** i les corresponents a l'etapa "FOTO 2-2" (les denominacions es remunten al capítol *Un nou camí i també una drecera*) amb ****V*V****, igual que des dels mètodes 3 i 4, hi hauríem d'incorporar sentències condicionals que encara complicarien més una funció ja massa llarga i accidentada. Pot ser que només en veieu un, d'accés explícit (que és el **VTV** subratllat), però també cal comptar-hi els quatre accessos a **ANORMALS**. I centrats ja en aquesta funció, no tant per la seva primera expressió **(if TRANSP (setq VTV **V*V** TRANSP ()))** (només hi és en atenció al moment en què el mètode 2 treu **FOTO2** i s'oblida de **TRANSP**) com per l'accés de la línia següent a **FOTO-V*V** (per treure aquesta foto): la funció **FOTO-V*V** utilitza **VTV** (i no ****V*V****) per tal que **FOTO1** respongui a la pseudomatriu 9×9 més eficient per a **PERFIL-V*V** i **COMPAT-PERFILS** (deteminada per **DET-TRANSP** des de la primera d'aquestes funcions), que mantindrem actualitzada fins que **ANORMALS** tregui **FOTO2**; amb el mètode 4, això obliga a inicialitzar **VTV** des de **FES-SUDOKU** i a localitzar aquí aquesta variable.

Amb aquest darrer aclariment, l'autor opina que ja ha dit tot el que havia de dir sobre la transposició de les pseudomatrius de treball als efectes d'optimització, optimització que hem acabat limitant (quan **DET-TRANSP** ens en dicta la pertinència) al mètode 2 i, dintre d'aquesta opció estratègica, a l'etapa anomenada "FOTO 1-2". Hem justificat per què deixàvem fora l'etapa "FOTO 2-2" (renunciant a influir en l'acció d'**ANORMALS**, en no tenir garanties que la influència sempre fos favorable) i també per què ens absteníem de tota intervenció quan optàvem pels mètodes 3 i 4. Però no volem tancar el tema sense donar al lector que no comparteixi tal decisió l'oportunitat de perseverar en la seva legítima discrepància (al cap i a la fi es tracta d'un terreny en què les certeses són molt subjectives i queda un marge prou ampli per a apreciacions divergents) i, fins i tot, facilitats per tirar endavant. Tampoc no cal sobrevalorar la generositat del gest, perquè l'oferiment d'un codi alternatiu en què la transposició segons el criteri **DET-TRANSP** s'aplica en totes les ocasions (per reduir la durada de **PERFIL-V*V**, **COMPAT-PERFILS** i **ANORMALS**, amb el mètode 2, i la d'**ANORMALS** amb els nous mètodes 3 i 4) gairebé no li ha suposat a l'autor cap dedicació extra: no fa massa parlàvem d'una versió *ad hoc* utilitzada precisament en l'exploració de més exemples, per tal de veure si la transposició i normalització de la VERSIÓ ESCOLLIDA, substituint la V. E. NORMALITZADA, suposava un estalvi de temps en **ANORMALS** com ho suposava en **PERFIL-V*V** i **COMPAT-PERFILS**, és a dir, sempre que la distribució d'emplenaments per requadres 9×3 fos més uniforme en la transposada que en l'original (hem subratllat la formulació del criteri que aplica **DET-TRANSP** perquè també havíem vist que, si hi ha la mateixa diferència de caselles emplenades entre el requadre 9×3 més poblat i el més despoblat que entre el requadre 3×9 més poblat i el més despoblat, pot passar qualsevol cosa), i només ha calgut polir una mica aquesta versió. Molt bé! -pensareu- però, ¿que no és hora de donar per tancat el tema? És clar, però posar el codi provisional, com farem de seguida, a disposició de qui se'n vulgui servir, no és reobrir el debat. Només és deixar una porta oberta als qui estiguin interessats a seguir esbrinant per què en alguns casos la transposició abreuja l'execució d'**ANORMALS** i a d'altres no, tasca en què serà útil disposar de les dues versions del programa per poder comparar les durades d'**ANORMALS** (si aquesta funció intervé en un moment en què la distribució d'emplenaments per requadres és més regular a la versió transposada, circumstància que es detecta fàcilment): quan la divergència fos significativa, la transposició en si no seria la responsable (ho és en l'etapa "FOTO 1-2", en què un mateix estat d'emplenament pot tenir un nombre diferent de solucions normalitzades compatibles, segons que considerem versió escollida original o transposada, però les simplement compatibles ja no depenen d'això) i caldria buscar la causa entre altres factors. Val a dir que parlem de diferències no gens menystenibles. Per exemple, quan dèiem que la intenció inicial havia estat precisament estendre a **ANORMALS** el dispositiu de transposició automàtica, perquè els primers resultats experimentals eren prou satisfactoris, ens referíem sobretot al primer sudoku mínim (17) de la col·lecció publicada per Gordon Royle, que gràcies al dispositiu havia passat a ser abordable amb el mètode 2, amb una espera de **5'25"** pel que fa a **PERFIL-V*V** i **COMPAT-PERFILS**, i una resposta immediata pel que fa a **ANORMALS**, però no havíem arribat a precisar que, sense la transposició, la durada d'**ANORMALS** passava de nul·la a ser de **3'35"** i que, adoptant els mètodes 3 o 4, la desactivació de l'antepenúltima, penúltima i última casella era instantània amb la transposició, i comportava esperes de **0'40"**, **6'20"** i **44'05"** sense ella. Tanmateix, quan anàvem al quart sudoku de la col·lecció (el segon s'assemblava massa al primer i en el tercer no es donaven les condicions que desencadenaven una transposició automàtica) la situació s'invertia: **PERFIL-V*V** i **COMPAT-PERFILS** eren despatxats en **0'50"**, l'execució d'**ANORMALS** durava **7'10"** amb transposició i era instantània sense ella, i quan adoptàvem els mètodes 3 o 4 les desactivacions de l'antepenúltima, penúltima i última casella comportaven esperes de **0'35"**, **1'40"** i **3'15"** amb la transposició, i esdevenien instantànies sense ella.

```
(defun N-3*3 (9*9 P-SEMPRE / 3*3 *3 I)
  (setq K -1)
  (repeat 3
    (setq *3 ())
    (repeat 3
      (setq J -1 K (1+ K))
      (repeat 3
        (setq N 0)
        (repeat 3
          (setq J (1+ J) I (nth J (nth K 9*9)))
          (if (or (atom I) (and P-SEMPRE (equal I P))) (setq N (1+ N))))
```

```
        (setq *3 (cons N *3))))
    (setq *3 (list (+ (nth 2 *3) (nth 5 *3) (nth 8 *3))
                   (+ (nth 1 *3) (nth 4 *3) (nth 7 *3))
                   (+ (nth 0 *3) (nth 3 *3) (nth 6 *3))) 3*3 (cons *3 3*3)))
  (setq 3*3 (reverse 3*3)))

(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP*)
  (setq N (N-3*3 (if 1-2 **V**V** **9*9**) T) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ))
  ..........................................................)

(defun VAR->NORM (/ K L 9**9 A B C D)
  (setq **9**9** (TRANSPOSAR **9**9**) K -1)
  (repeat 3
    (setq L ())
    (repeat 3 (setq K (1+ K) L (cons (nth K **9**9**) L)))
    (ORDENA-3)
    (foreach E L (if E (setq L (list A E D))))
    (setq 9**9 (append 9**9 (list L))))
  (setq K -1 L ())
  (foreach E 9**9 (setq L (cons (cons (caar E) E) L)))
  (ORDENA-3)
  (foreach E L (if E (setq **9*9** (append (cdr A) (cdr E) (cdr D)))))
  (if (not T**9*9**)
      (progn
        (setq T**9*9** **9*9**)
        (if (= 1-2 "4") (setq T**V*V** **9*9**)))))

(defun PRE-NORM->VAR (/ L* L** N F F1 F2 F3 F4)
  (foreach L  (reverse **9*9**) (setq L*  (cons (car L) L*)))
  (foreach L (reverse **9**9**) (setq L** (cons (car L) L**)))
  (setq N1 (- 9 (length (member (nth 0 L*) L**))) F1 (/ N1 3)
        N2 (- 9 (length (member (nth 3 L*) L**))) F2 (/ N2 3)
        ..........................................................
        F3 (- 9 (length (member (nth 8 L*) L**))) F3 (rem F3 3)
        N3 (list F1 F2 F3) N3 (PERMUT-N N3))
  (if (not TN1) (setq TN1 N1 TN2 N2 TN3 N3 TN4 N4)))

(defun NORMTEXT-9*9 (/ T**9*9**)
  (prompt "\n\n\nSi anomenem SOLUCIÓ NORMALITZADA aquella en què les files de ")
  ....................................
  (prompt " ja queda determinada.")
  (SEGUIR ())
  (repeat 2 (VAR->NORM) (PRE-NORM->VAR))
  (graphscr)
  ........................................
  (prompt "i després seguir com si no res.")
  (SEGUIR ()))

(defun TV->TN (N1 N2 N3 N4 / N)
  (cond ((< Y 3) (Y1 3 2 3 4 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 4 1 3))
        (T (Y3 3 1 4 3 5)))
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 3 4 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 4 1 3))
        (T (Y3 1 1 4 3 5))))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if (and (> 1-2 "1") Q) (setq P Q X (car P) Y (cadr P) PX QX PY QY))
        ..............
        (if (> 1-2 "1")
            (progn
              (command "CVPORT" 2)
              ..................
              (command "CVPORT" 3))))
```

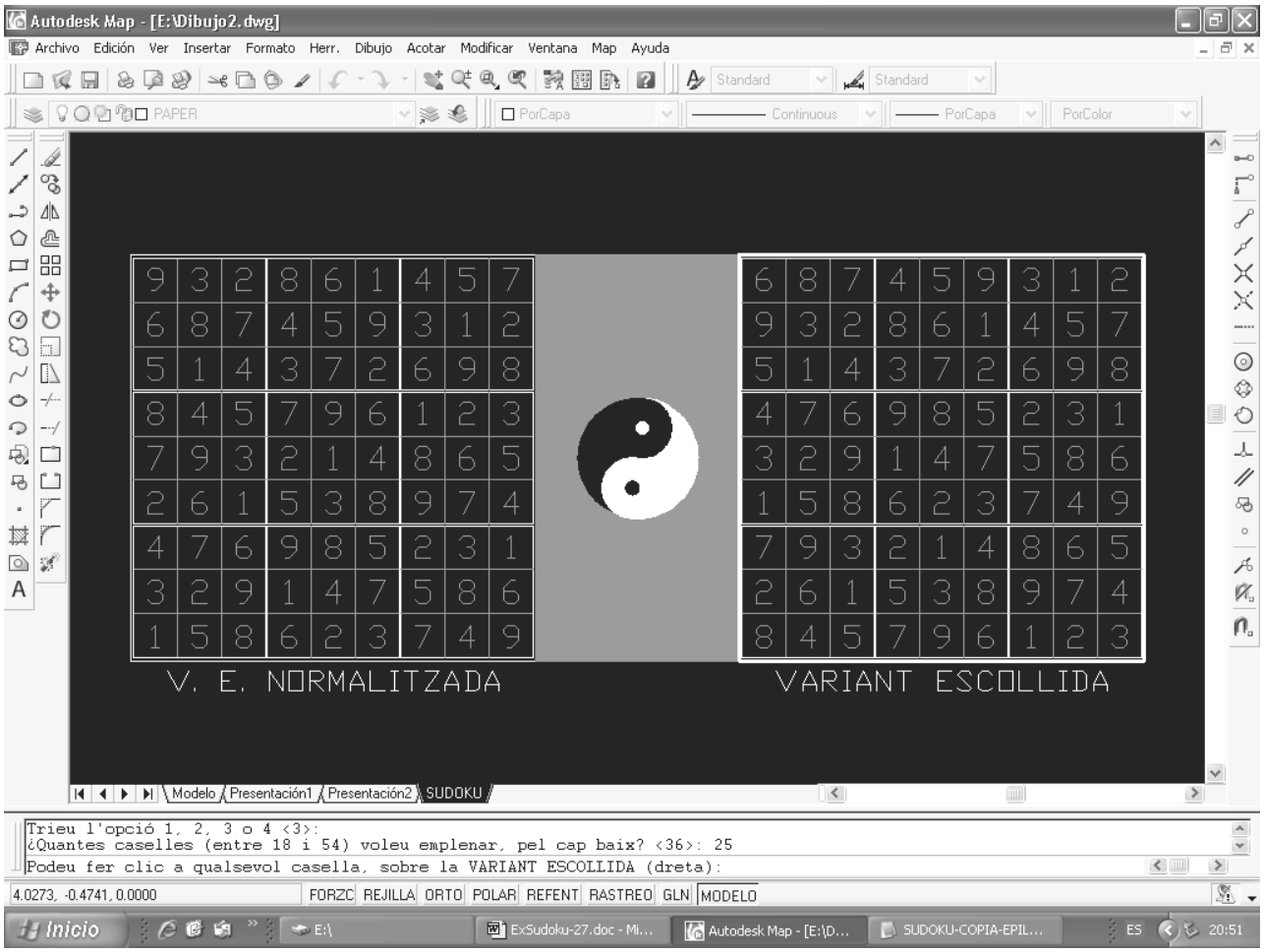
```
      (progn
        (setq N (ELEMENT **9*9**) ...)
        ..............................
        (setq P Q X (car P) Y (cadr P) PX QX PY QY Q ())))
  (if (> 1-2 "1")
      (progn
        (setq TP (reverse (if (> (getvar "CVPORT") 2) Q R))
              X (car TP) Y (cadr TP))
        (TV->TN TN1 TN2 TN3 TN4)
        (setq T**V*V** (COMMUTA T**V*V**) TP (list X Y)
              VTV (if TRANSP T**V*V** **V*V**)
              PT (if TRANSP TP P)))))

(defun FOTO-V*V (/ FOTO)
  (setq J -1)
  (foreach L VTV
    (setq I -1 J (1+ J))
    (foreach E L
      (setq I (1+ I))
      (if (atom E) (setq FOTO (cons (list I J) FOTO)))))
  FOTO)

(defun DET-TRANSP (/ N Ñ)
  (setq N (N-3*3 **V*V** ()) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ) ; També es pot fer amb **V**V**.
        TRANSP (> (- (eval (cons 'max N)) (eval (cons 'min N)))
                  (- (eval (cons 'max Ñ)) (eval (cons 'min Ñ))))
        VTV (if TRANSP T**V*V** **V*V**) PT (if TRANSP TP P)))

(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V KLINIA-V*V KKE KE)
  (if (not L1) (C:CARREGA-123456789))
  (DET-TRANSP)
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
  (repeat 3
    (setq J (- J 9) K (+ K 3))
    (repeat 3
      (setq J (+ J 3) K (- K 3) W ())
      (repeat 3
        (setq J (- J 3) K (1+ K))
        (repeat 3
          (setq J (1+ J) V (nth J (nth K VTV)))
          (if (atom V) (setq W (cons V W)))))
      (cond ((= J 2) (setq A W))
            ((= J 5) (setq B W))
            (T       (setq C W))))
    (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
  (setq L1C (reverse L1C) J -1)
  (repeat 9
    (setq J (1+ J) K -1 L ())
    (repeat 9 (setq K (1+ K) C (nth J (nth K VTV))
                    L (if (atom C) (cons C L) L)))
    (setq L2C (cons L L2C)))
  (setq L2C (reverse L2C) LINIES-V*V () KLINIES-V*V () K -1)
  (terpri)
  (repeat 9
    (setq LINIA-V*V () KLINIA-V*V () L "" K (1+ K))
    (mapcar '(lambda (E F G)
               (setq L (strcat L (if (or F G)
                                     (if (atom E)
                                         (itoa E)
                                         (progn
                                           (setq C "]")
                                           (foreach H (append F G)
                                             (setq C (strcat (itoa H) C)))
                                           (strcat "[~" C)))
                                     "#"))))
            (nth K VTV) (nth K L1C) L2C)
    ..........................................................................)
```

```
  (setq LINIES-V*V (reverse LINIES-V*V)
        KLINIES-V*V (reverse KLINIES-V*V)
        L1 () *KK* (reverse *KK*))
  (terpri))

(defun ACLARIMENT ()
  (terpri) (***)
  (prompt "\n Per tal d'abreujar aquest procés, s'ha substituït la VERSIÓ")
  (prompt " ESCOLLIDA per la\n seva transposada (substitució de files per")
  (prompt " columnes), però com que seria poc\n entenedor al•ludir a \"")
  (prompt "solucions normalitzades compatibles amb el resultat de\n normalitzar")
  (prompt " la transposició de la VERSIÓ ESCOLLIDA\", no s'ha canviat el text.")
  (***)
  (prompt "\nA partir d'ara, l'expressió \"solucions normalitzades\"")
  (prompt " s'haurà d'entendre així.")
  (***) (terpri))

(defun COMPAT-PERFILS (/ L1 LL0 KLL0 LL1 KLL1 L2 LL2 KLL2 KT01 3L M N TERCETS-V*V
                         CC0 PCC0 CC1 PCC01 CC2 CCC1 CCC2 TT0 TT1 TT2 LOCT1 LOCT2
                         TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  ......................................
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq KLINIES-V*V () TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        ......................................................................
        (setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
              KK (length S-V*V) S-V*V (list (cons PT S-V*V)))
        (prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                                       (nth 1 *TT*) " tríades 4-5-6 i les "
                                       (nth 2 *TT*) " tríades 7-8-9,"
                        "\nde solucions normalitzades compatibles amb la V. E. "
                        "NORMALITZADA n'hi ha " (itoa KK) "."))
        (textscr)
        (if TRANSP (ACLARIMENT))
        (SEGUIR ())
        (graphscr))))

(defun ANORMALS (NORM=1 1-2<34 / L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P PRE
                                 NORM)
  (DET-TRANSP)
  (setq L0A8 '(0 1 2 3 4 5 6 7 8) FOTO1 () FOTO2 (FOTO-V*V))
  (if (not NORM=1) (setq KKK () SUDOKUS-V*V ()))
  (foreach X L0A8
    (setq XX ())
    (foreach Y L0A8 (if (atom (setq V (ELEMENT VTV))) (setq XX (cons V XX))))
    (setq XXX (cons XX XXX)))
  (setq XXX (reverse XXX))
  (foreach Y L0A8
    (setq YY ())
    (foreach X L0A8 (if (atom (setq V (ELEMENT VTV))) (setq YY (cons V YY))))
    (setq YYY (cons YY YYY)))
  (setq YYY (reverse YYY))
  (foreach YY '(0 3 6)
    (foreach XX '(0 3 6)
      (setq L () Y -1)
      (foreach VV VTV
        (setq Y (1+ Y) X -1)
        (if (not (or (< Y YY) (> Y (+ YY 2))))
            (progn
              (foreach V VV
                (setq X (1+ X))
                (if (not (or (< X XX) (> X (+ XX 2))))
                    (if (atom V) (setq L (cons V L))))))))
      (setq XY (cons L XY))))
  (setq XY (reverse XY))
```

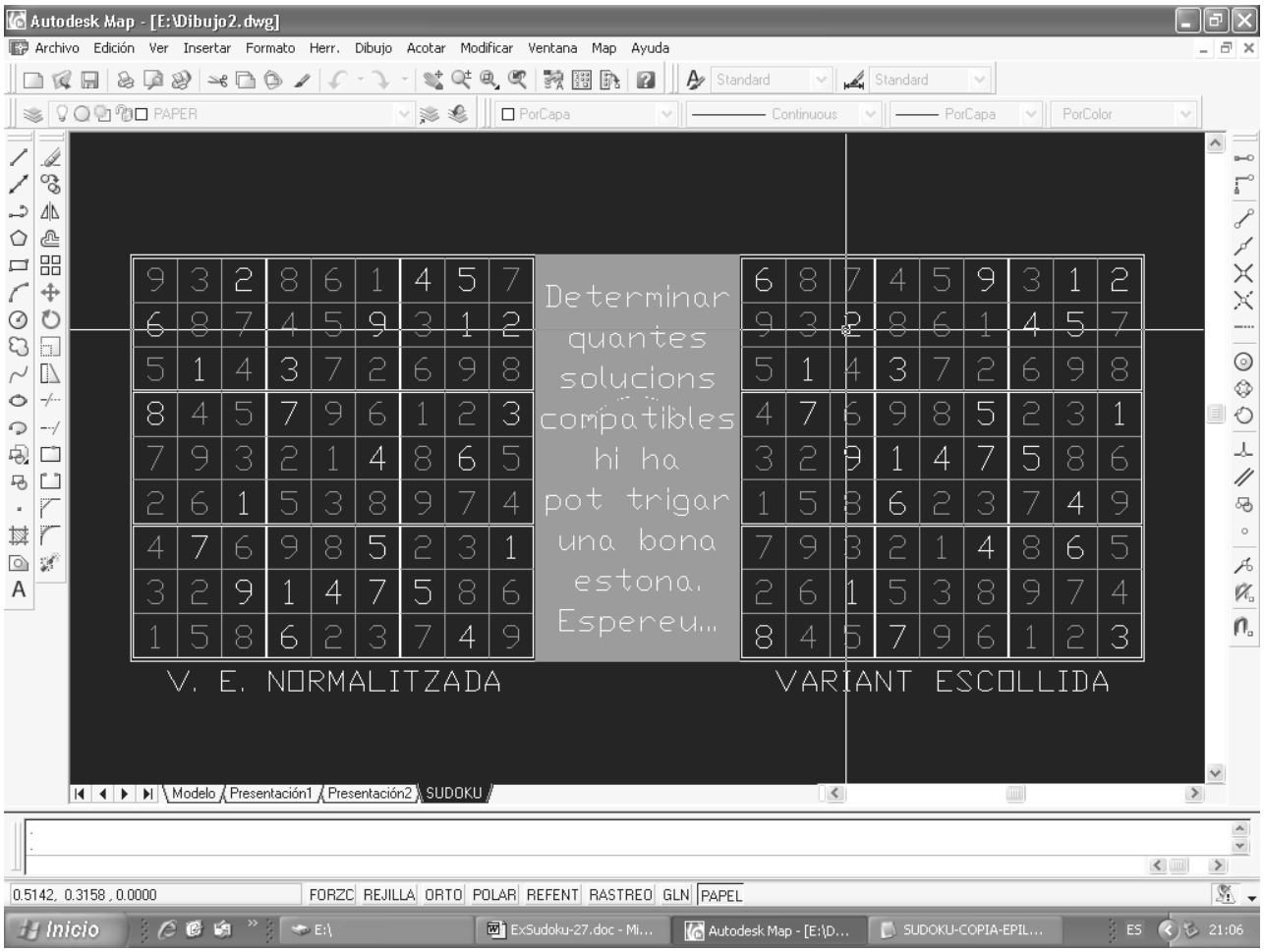
```
  (foreach Y L0A8
    (setq LL ())
    (foreach X L0A8
      (setq L ())
      (if (listp (ELEMENT VTV))
          (foreach E L1A9
            (setq L (cons (if (or (member E (nth X XXX)) (member E (nth Y YYY))
                                  (member E (nth (+ (/ X 3) (* (/ Y 3) 3)) XY)))
                              ()
                              E)
                          L))))
      (setq LL (cons (reverse L) LL)))
    (setq LL (reverse LL) LLL (cons LL LLL)))
  (setq LLL (reverse LLL))
  (if NORM=1
      (progn
        (setq PRE T LL ())
        (foreach EE (TRANSPOSAR VTV)
          (setq L ())
          (foreach E EE (setq L (cons (if (atom E) E (reverse E)) L)))
          (setq LL (cons (reverse L) LL)))
        (setq LL (reverse LL))
        (RE+MEMBER LL (TRANSPOSAR LLL)))
      (RE+MEMBER VTV LLL))
  (setq KK (length KKK) KKK (cons PT KKK) S-V*V (list KKK))
  (if (and NORM=1 (= KK 1)) (setq KKK ())))

(defun FES-SUDOKU-234 (/ LINIES-V*V KLINIES-V*V *K* *KK* R RX RY TP PT)
  (setq Q ())
  (CLIC)
  (if POST
      (if V
          (if (or (member PT FOTO1) (member PT FOTO2))
              (if (= 1-2 "4")
                  (setq POST ())
                  (progn
                    (command "ESPACIOP" "BORRA" "LT" ""
                             "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                    (RETOL-2 "Anul-lant" "aquesta" "casella," "caldrà"
                             "recalcular" "solucions" "compatibles"
                             "i tornar a" "esperar...")
                    (if FOTO2
                        (progn
                          (SEGUIR T)
                          (command "DESHACER" "R" "DESHACER" "M"
                                   "INSERT" "C" "0,0" "" "" "")
                          (RETOL-2 "...i, si surt" "\"EL SUDOKU" "JA TÉ UNA"
                                   "SOLUCIÓ" "ÚNICA\", mai" "no sabreu"
                                   "si sobren" "caselles" "plenes.")))
                    (initget "Si No")
                    (setq CONF
                          (getkword "\n.\n.\nConfirmeu l'anul·lació (S/N)? <N>: ")
                          CONF (if CONF CONF "No"))
                    (if (= CONF "Si")
                        (setq POST ())
                        (progn
                          (command "DESHACER" "R" "UY" "ESPACIOM")
                          (CLIC))))))
              (progn
                (setq I (assoc PT S-V*V) S-V*V (SUPR I S-V*V)
                      I (caar S-V*V) J (nth (car I) (nth (cadr I) VTV)))
                (NUMCOMPAT I J ())))
          (progn
            (command "ESPACIOP" "DESHACER" "M"
                     "INSERT" "C" "0,0" "" "" "")
            (RETOL-234)
            (NUMCOMPAT PT N T)
            (command "DESHACER" "R"))))
```

```
(if (not POST)
    (progn
      (if (and V (or (member PT FOTO1) (member PT FOTO2)))
          (setq POST T)
          (if (not V) (SUD-MIN)))
      (if POST
          (progn
            (if V
                (if (< 1-2 "4")
                    (command "DESHACER" "R")
                    (command "ESPACIOP"))
                (progn
                  (command "ESPACIOP")
                  (if (= 1-2 "2") (command "BORRA" "LT" ""))))
            (if (or (not (or FOTO1 FOTO2)) (member PT FOTO1))
                (if (= 1-2 "2")
                    (progn
                      (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                      (RETOL-2 "Determinar" "quantes" "solucions" "normalitza-"
                               "des són" "compatibles" "pot trigar" "hores."
                               "Espereu...")
                      (PERFIL-V*V)
                      (COMPAT-PERFILS)
                      (command "DESHACER" "R" "UY" "ESPACIOM"))
                    (progn
                      (prompt "\n.\n.\n")
                      (command "INSERT" "C" "0,0" "" "" ""
                               "DESHACER" "M")
                      (RETOL-234)
                      (ANORMALS () T)
                      (command "DESHACER" "R")
                      (if (= KK 1)
                          (progn
                            (while
                              (and (= KK 1) (> (length PPM) 1))
                              (prompt "Només hi ha una solució compatible i, ")
                              (prompt "just per això, cal mirar\nquantes n'hi")
                              (prompt " hauria amb una casella menys (")
                              (prompt (itoa (setq NUM (1- NUM))))
                              (prompt " caselles plenes).")
                              (SEGUIR ())
                              (command "ESPACIOM")
                              (setq Q (car PPM) PPM (cdr PPM)
                                    Q (if (< (getvar "CVPORT") 3)
                                          (car Q)
                                          (cadr Q))
                                    P (car Q) X (car P) Y (cadr P)
                                    PX (cadr Q) PY (last Q) Q ())
                              (CLIC)
                              (command "ESPACIOP")
                              (if (> (length PPM) 0)
                                  (progn
                                    (prompt "\n.\n.\n")
                                    (command "DESHACER" "M")
                                    (RETOL-234)
                                    (ANORMALS () ())
                                    (command "DESHACER" "R"))))
                            (if (> KK 1)
                                (progn
                                  (prompt "Us heu passat de la ratlla: ")
                                  (prompt "amb més d'una solució compatible, ")
                                  (prompt "el sudoku\nhaurà de recuperar ")
                                  (prompt "l'última casella esborrada ("
                                  (prompt (itoa (setq NUM (1+ NUM))))
                                  (prompt " caselles plenes).")
                                  (SEGUIR ())
                                  (command "ESPACIOM")
                                  (CLIC)
```

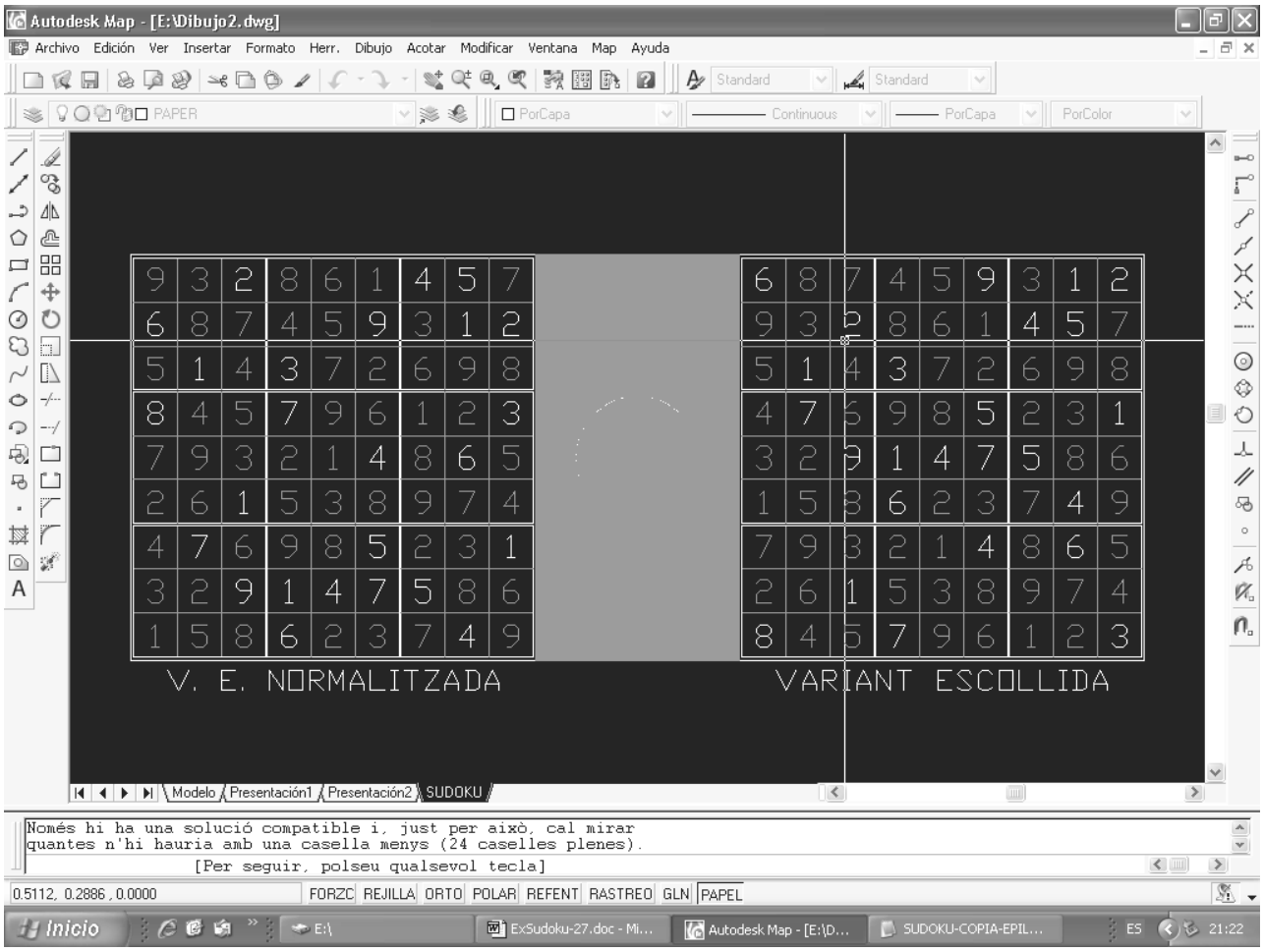
```
                                    (command "ESPACIOP")
                                    (NUMCOMPAT PT N T))
                                  (prompt (strcat "\n.\nHeu tingut el privilegi "
                                                  "de crear un SUDOKU MÍNIM (no"
                                                  "més 17 caselles plenes).")))))
                        (command "BORRA" "LT" "")))
                    (progn
                      (command "DESHACER" "M"
                               "INSERT" "C" "0,0" "" "" "")
                      (RETOL-234)
                      (ANORMALS () (< 1-2 "4"))
                      (command "DESHACER" "R")
                      (if (< 1-2 "4") (command "UY" "ESPACIOM")))))))
  (if POST
      (progn
        (setq GR (strcat "\nEL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA.\nOmplint més case"
                         "lles el fareu més fàcil (<Intro> o < > per acabar)"))
        (if (or (and (= 1-2 "4") (= KK 1) (> (setq NUM (length FOTO2)) 17))
                (and (< 1-2 "4") (> KK 1)))
            (if (= 1-2 "4")
                (prompt (strcat "\n.\nAmb només una solució compatible, heu de "
                                "buidar més caselles:"))
                (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                                (if KKK "" "normalitzades ")
                                "heu d'omplir més caselles:")))
            (if FOTO2
                (progn
                  (if (= 1-2 "4")
                      (progn
                        (if (> KK 1)
                            (progn
                              (command "ESPACIOP" "DESHACER" "M"
                                       "INSERT" "C" "0,0" "" "" "")
                              (prompt "\n.\nUs heu passat de la ratlla: amb més ")
                              (prompt "d'una solució compatible, el sudoku\n")
                              (prompt "haurà de recuperar l'última casella ")
                              (prompt (strcat "esborrada (" (itoa NUM)))
                              (prompt " caselles plenes).")
                              (SEGUIR ())
                              (command "DESHACER" "R" "ESPACIOM")
                              (CLIC)
                              (command "ESPACIOP" "DESHACER" "M"
                                       "INSERT" "C" "0,0" "" "" "")
                              (NUMCOMPAT PT N T)
                              (command "DESHACER" "R"))
                            (prompt "\n.\nHeu tingut el privilegi de crear un "
                                    "SUDOKU MÍNIM (només 17 caselles plenes)."))
                        (setq 1-2 "3")))
                  (if KKK (setq FOTO2 (FOTO-V*V) KKK ()))
                  (prompt ...........................))
                (progn ..................................................))))))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     T**V*V** **V**V** VTV TRANSP)
  (if (= 1-2 "1") ...)
  (setq .............)
  (prompt (strcat "\n.\n.\n" GR))
  (if (= 1-2 "4")
      (setq VTV **V*V** FOTO2 (FOTO-V*V) POST T)
      (if (> 1-2 "1")
          (foreach EE (reverse (TRANSPOSAR **V*V**))
            (setq J ())
            (foreach E EE (setq J (cons (reverse E) J)))
            (setq T**V*V** (cons (reverse J) T**V*V**)))))
  (repeat 2 ........................................)
  ..................................................
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" ""))
```



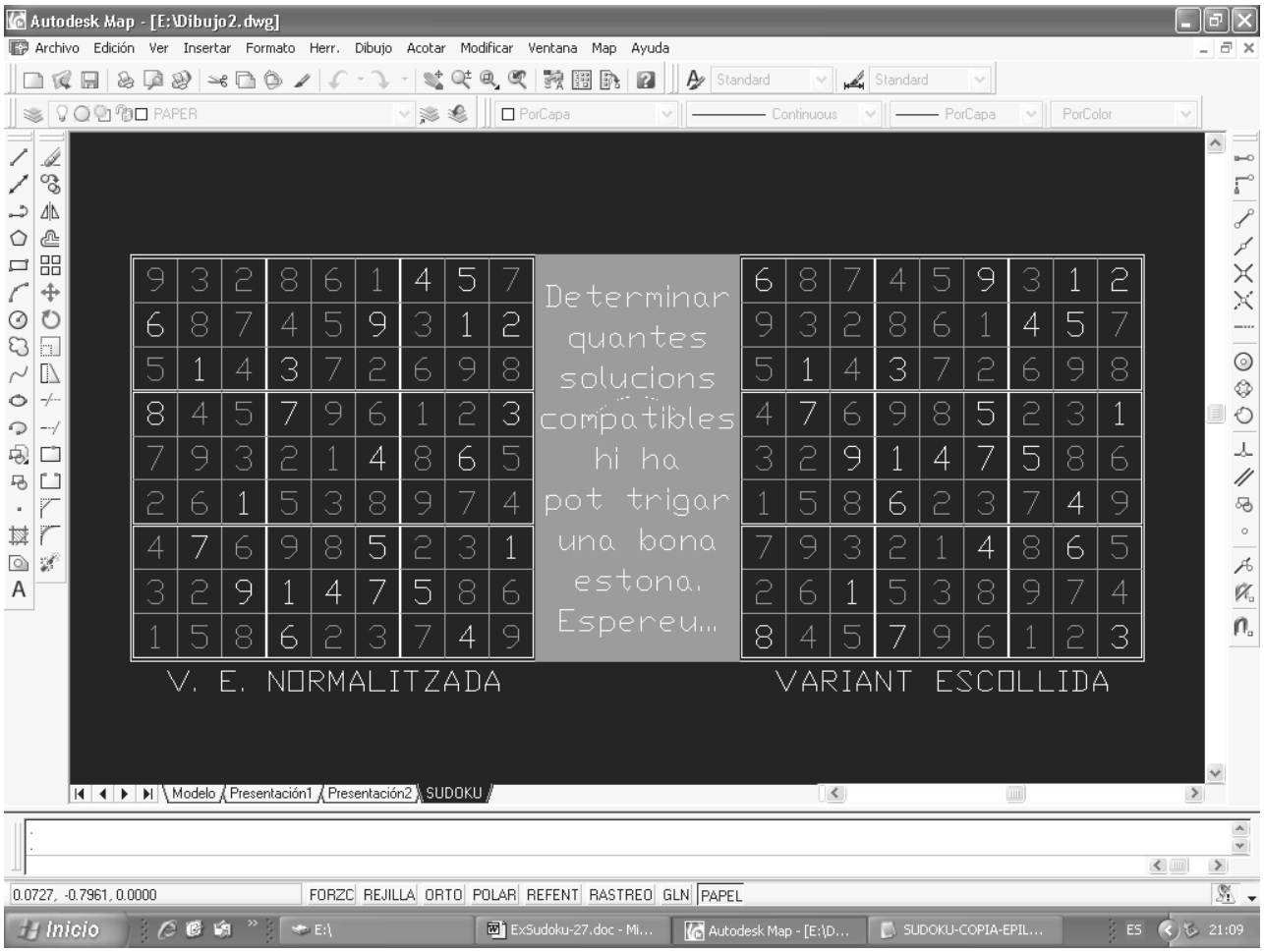
Amb el mètode 3 s'escull la quantitat de caselles activades a partir de la qual...



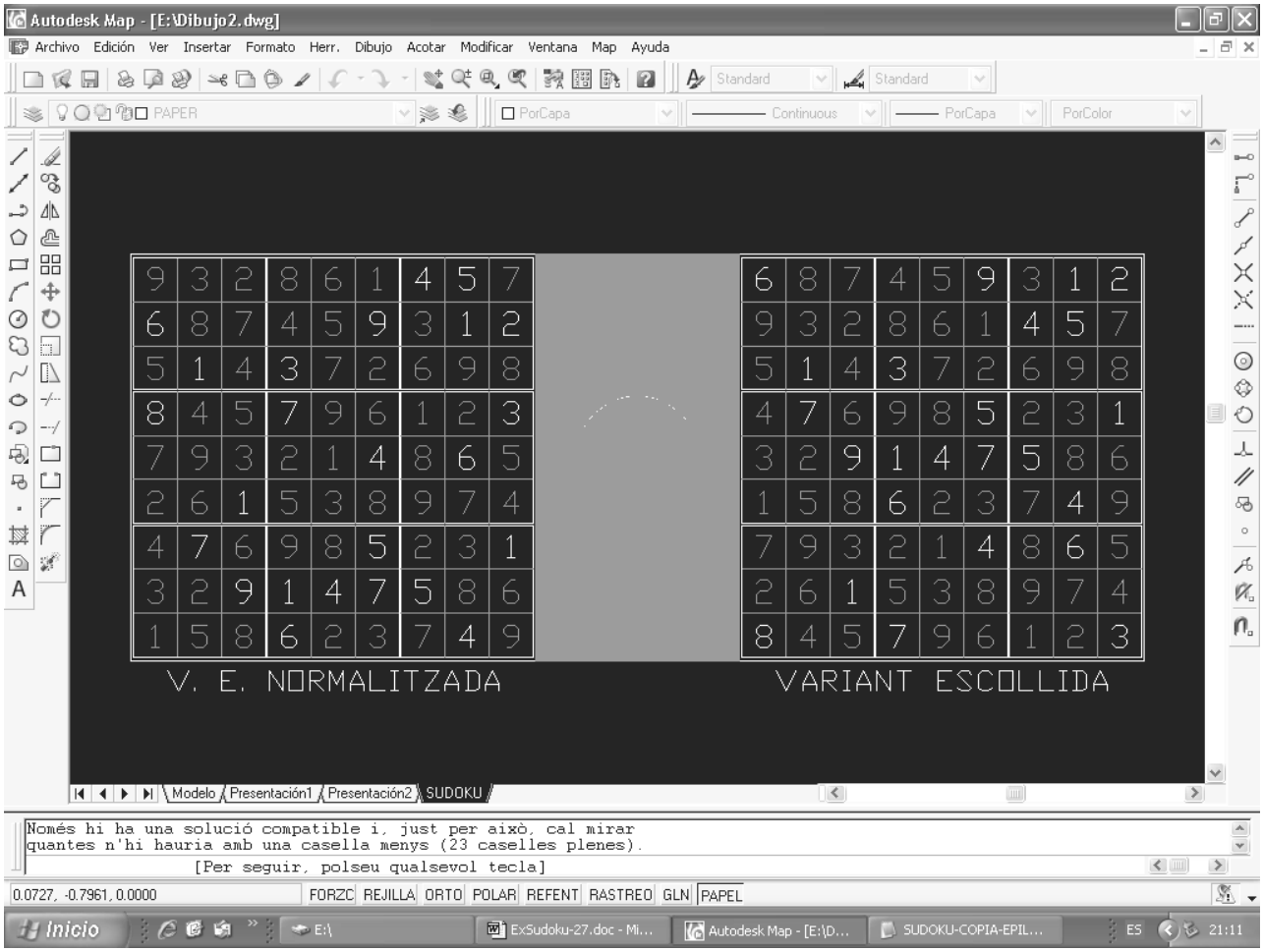
comença el recompte de solucions compatibles. Si aquest sudoku està massa ple...



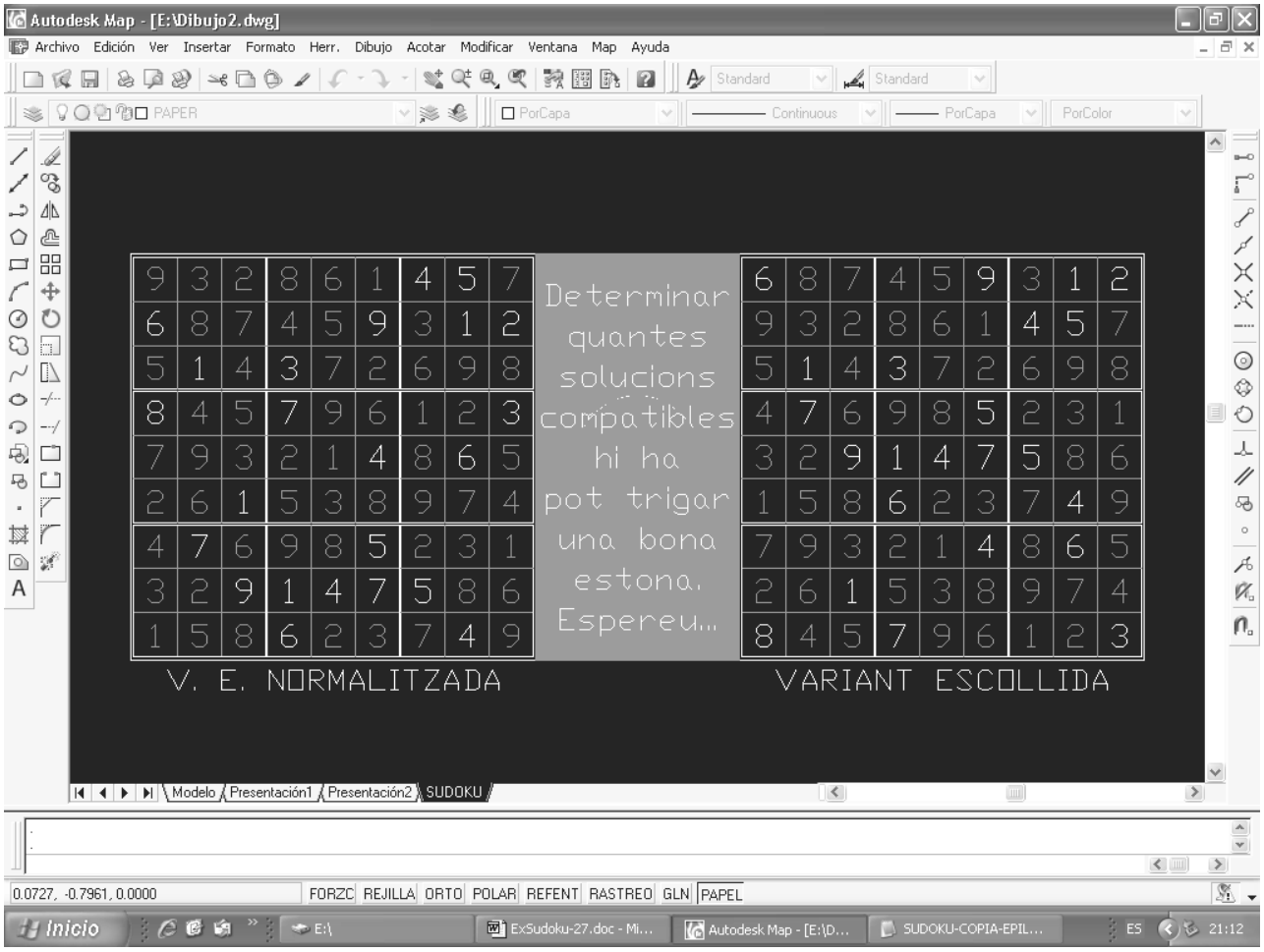
(millor això que massa buit), caldrà anar desactivant caselles en un ordre...



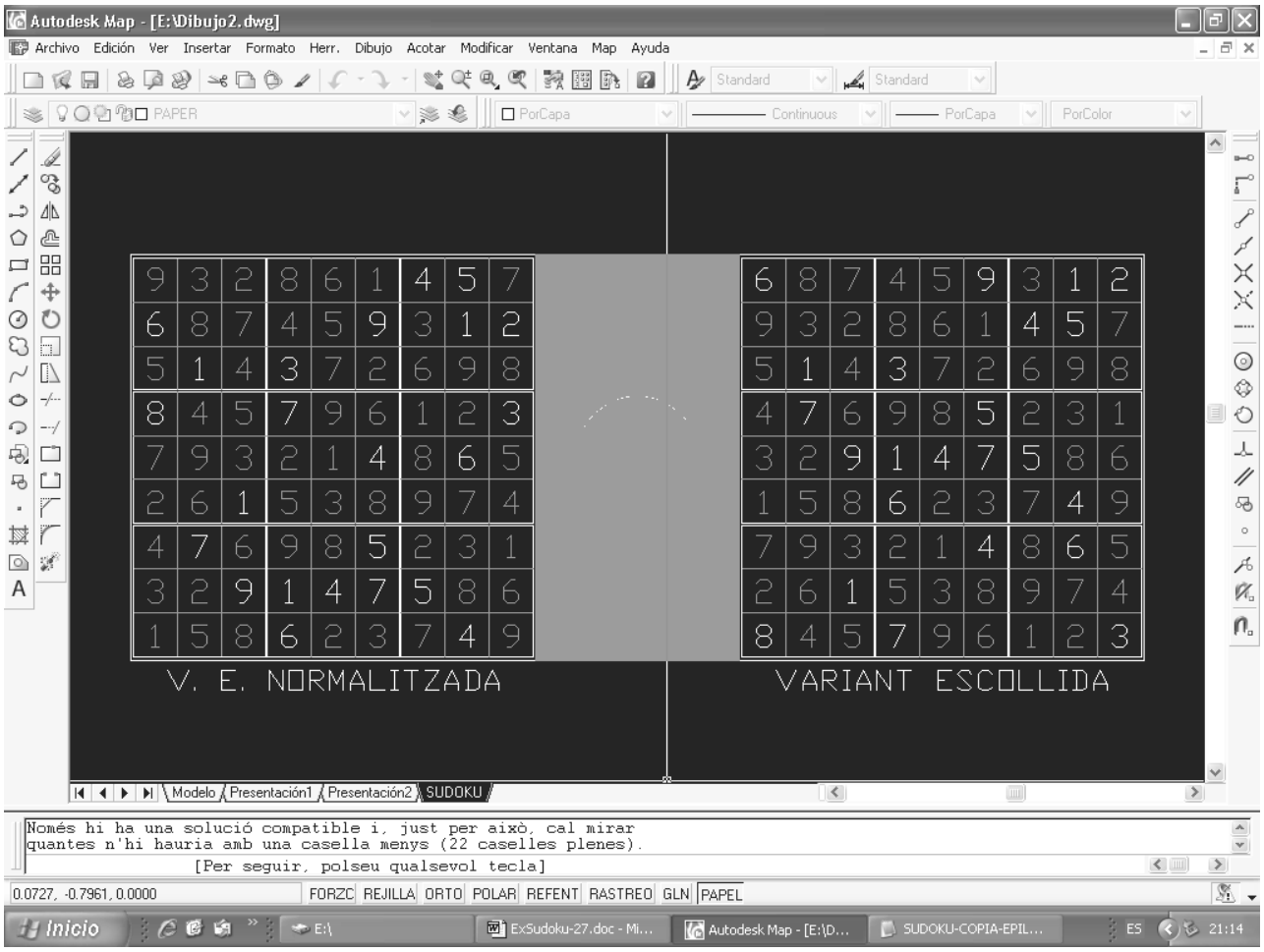
invers al d'activació, fins que quedi una única solució compatible i no poguem...



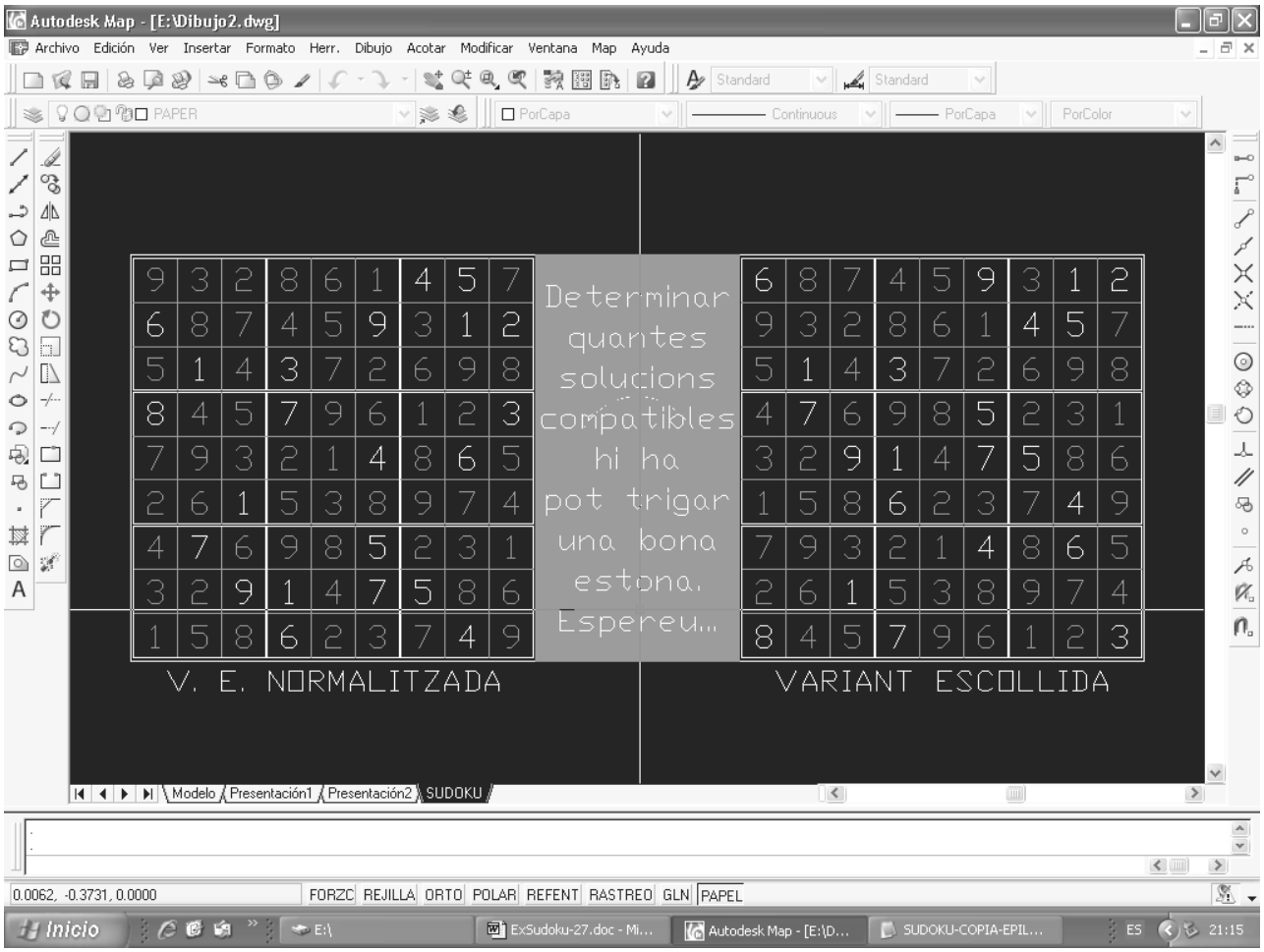
seguir desactivant (perquè així passaríem a tenir més d'una solució compatible).



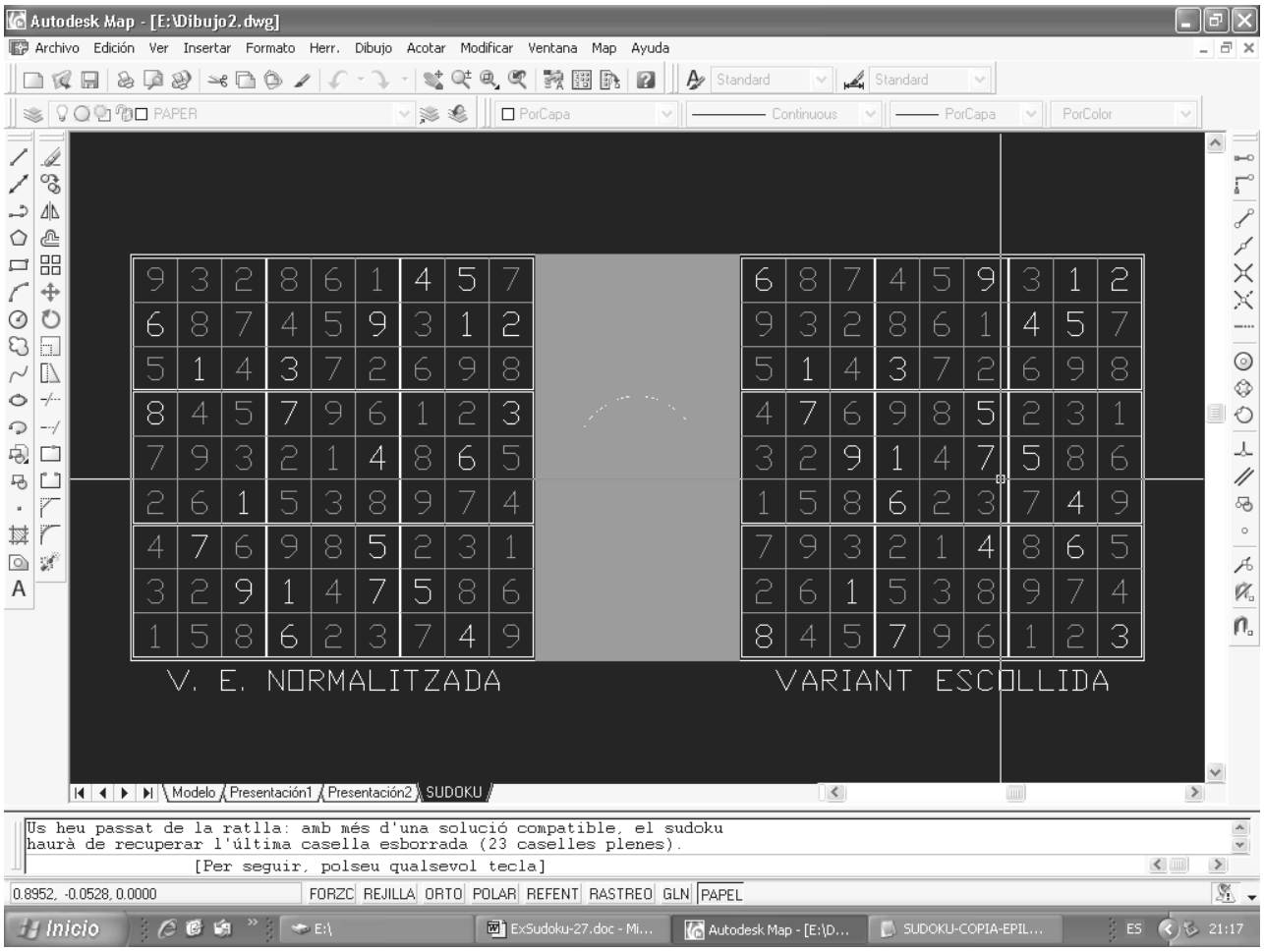
Dues precisions: mentre només hi hagi una solució compatible, el rètol central...



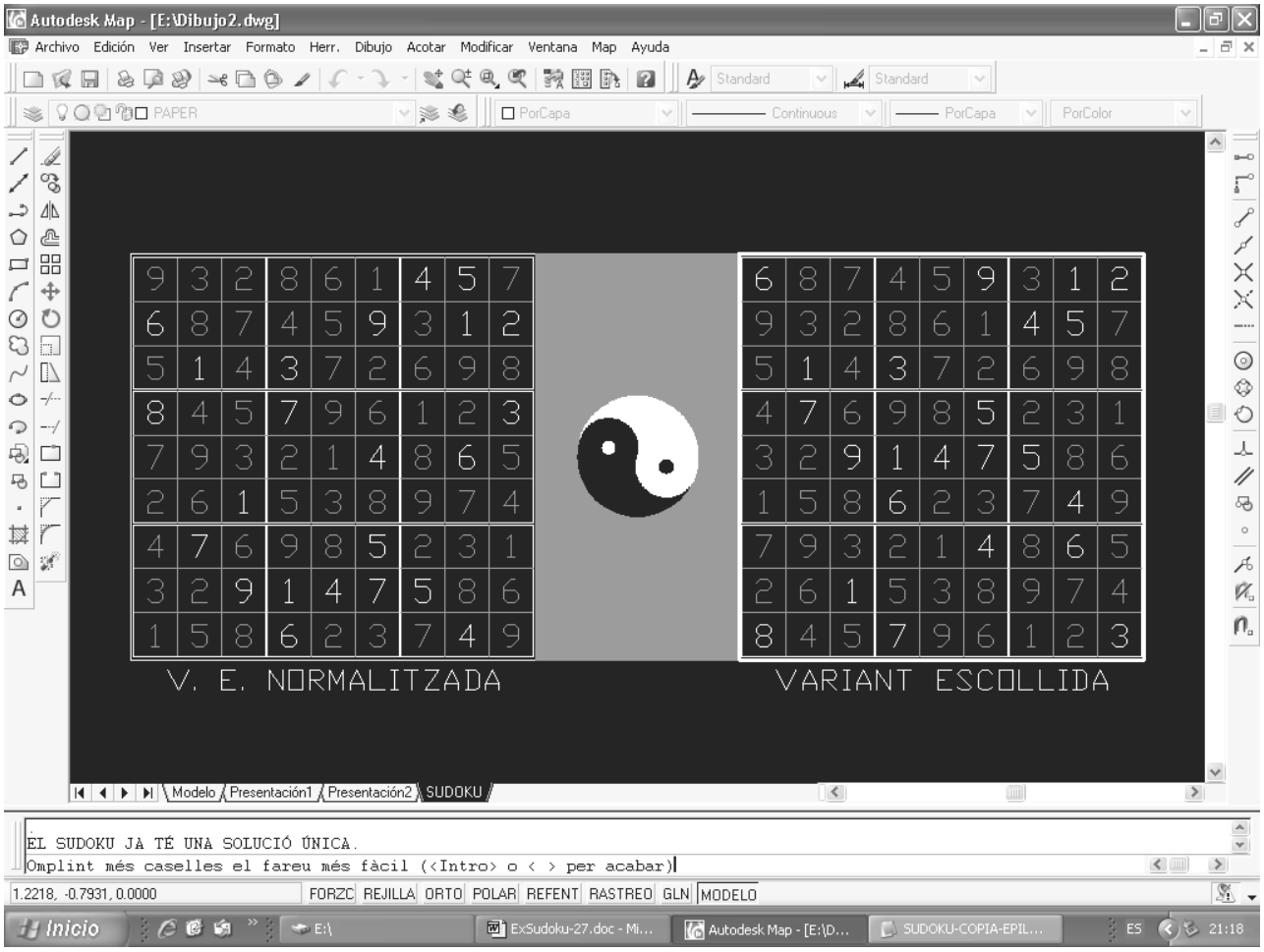
es veurà molt poc temps (s'ha hagut d'interrompre el procés per treure aquestes...



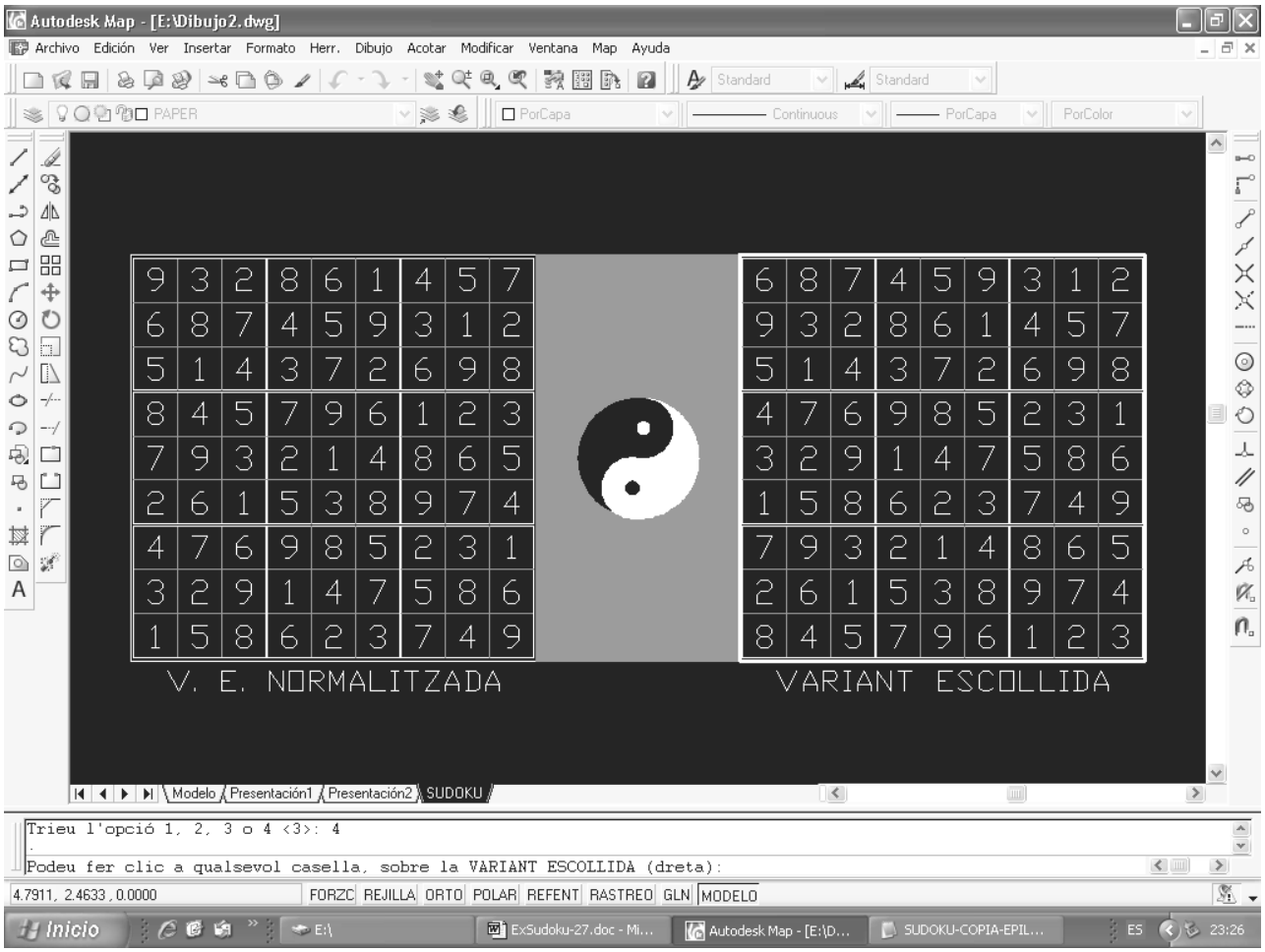
instantànies), de manera que això d'"una bona estona" només s'escau amb nivells...



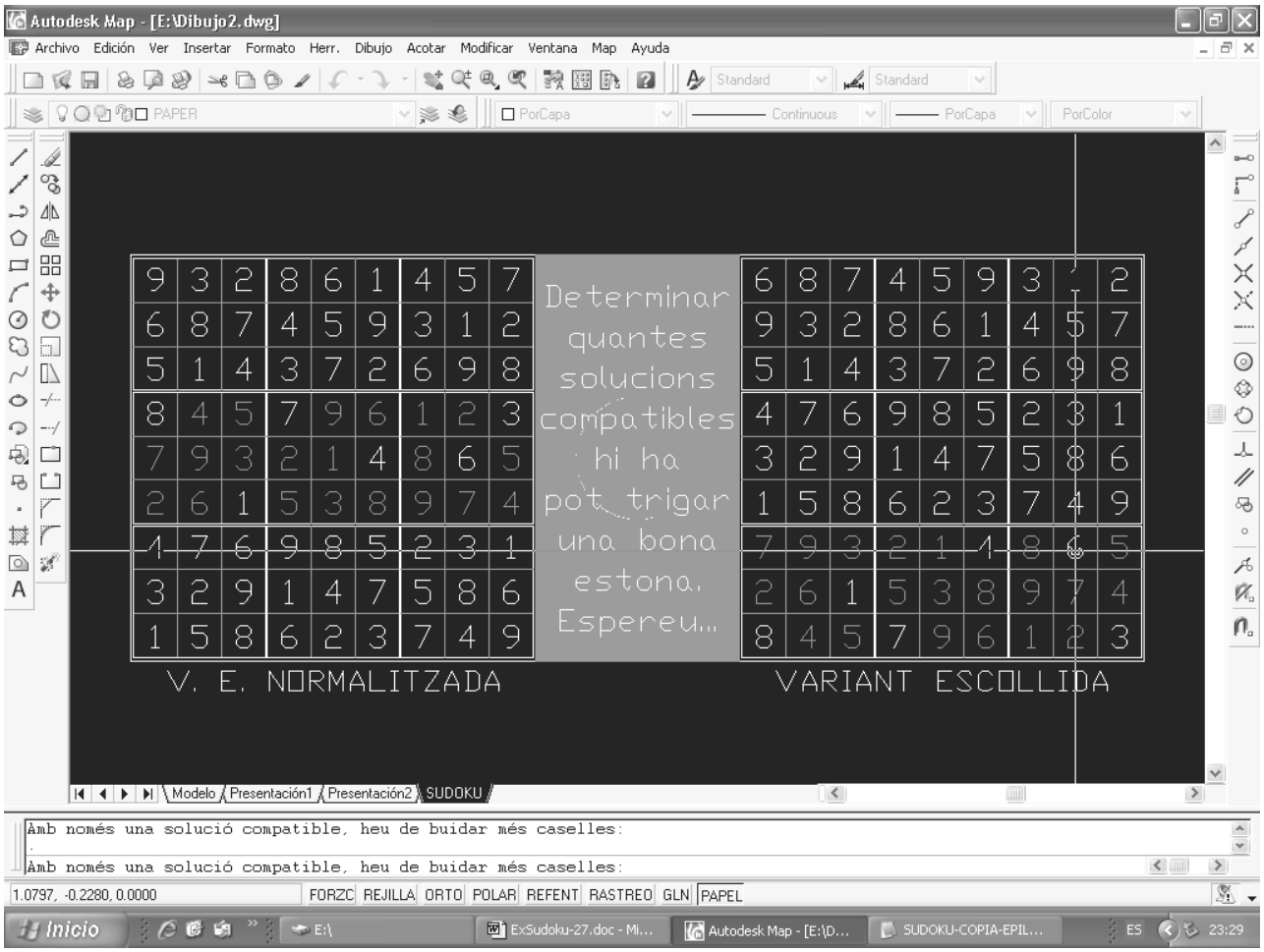
d'ocupació baixos; la segona és que mai no sabrem què pot passar amb una altra...



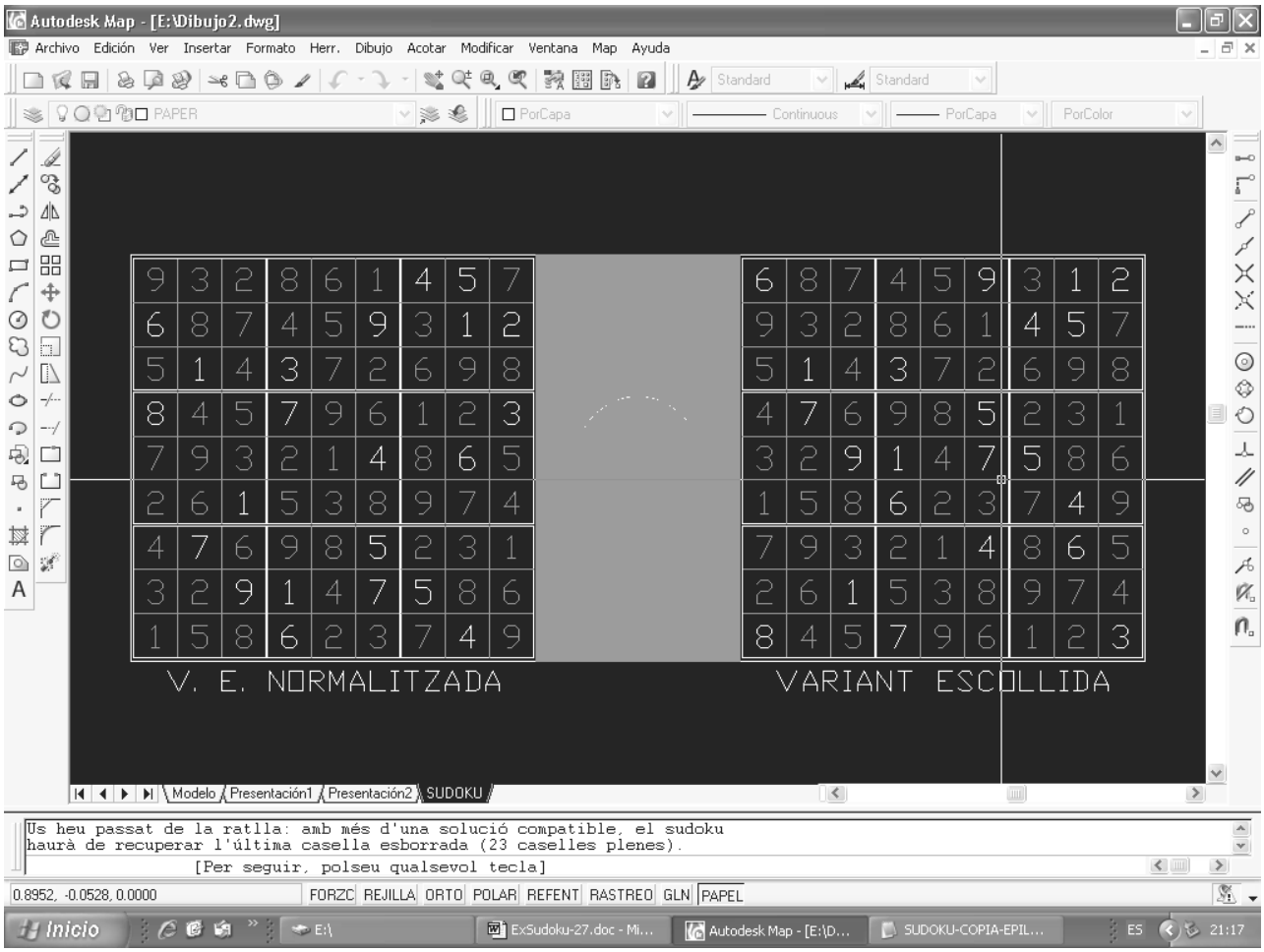
desactivació fins que no ho provem (només en cas de 17 ocupacions no caldrà).



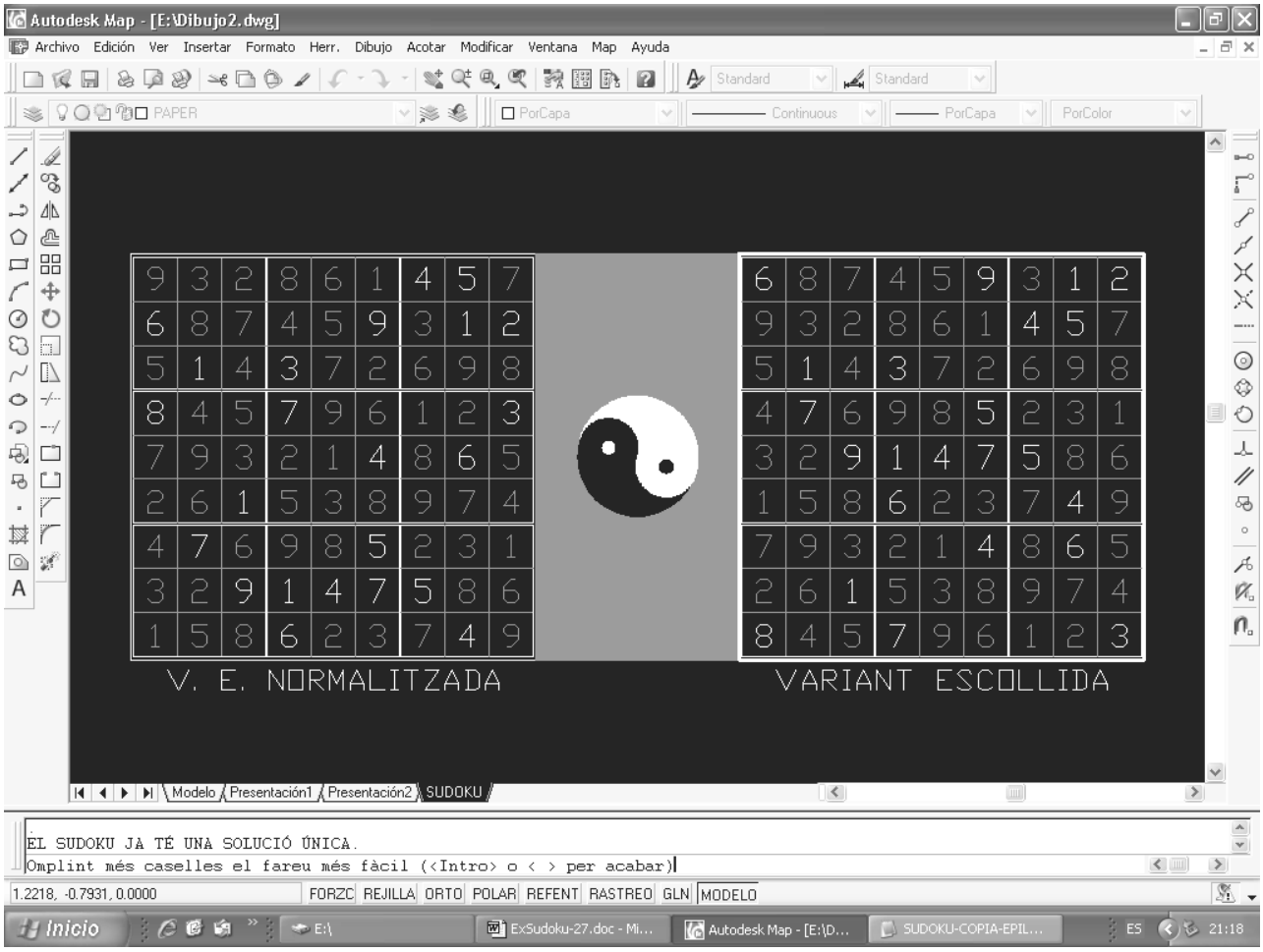
Amb el mètode 4 es parteix de les 81 caselles activades i es desactiven les que...



no volem deixar (en la instantània, hem fet el buidatge del primer requadre 9×3).



Igual que en el mètode 3 (tret dels sudokus de 17), sempre haurem de desactivar...



una casella més, desactivació que serà anul·lada (en l'exemple, la casella 5,5).

En haver arribat al final d'un capítol que havia de posar fi a tot el document, les darreres troballes realitzades han fet més aconsellable ajornar aquesta fita: faltant uns retocs que no comportaran innovacions d'estratègia ni de procediment, i que només pretenen servir el producte més ben acabat, serà millor traslladar-los a *Epíleg: això acaba igual que el conte de la sopa de còdols?*, afegitó no previst inicialment però que ens evitarà farcir encara més el capítol actual, concebut amb vocació de colofó i que tanmateix ha anat acumulant aportacions molt heterogènies.

Aquest cop de timó no obstarà per complir allò que havíem anunciat pàgines enrera, aprofitant la pausa: la demostració que, en el conjunt de tríades de valors enters positius $\mathbf{a}$, $\mathbf{b}$ i $\mathbf{c}$, de suma constant $\mathbf{a+b+c=k}$, el producte $\mathbf{a\,b\,c}$ és màxim quan $\mathbf{a=b=c=\frac{k}{3}}$. Val a dir que l'autor no pretenia demostrar només això sinó també que, quan més dispersos fossin els valors $\mathbf{a}$, $\mathbf{b}$ i $\mathbf{c}$, menor seria el producte $\mathbf{a\,b\,c}$, és a dir que, si tenim dues tríades $\mathbf{a_1+b_1+c_1=k=a_2+b_2+c_2}$ en què suposem els elements designats d'acord amb una ordenació com $\mathbf{a_1\le b_1\le c_1}$ i $\mathbf{a_2\le b_2\le c_2}$, $\mathbf{c_1-a_1<c_2-a_2\ \Rightarrow\ a_1\,b_1\,c_1>a_2\,b_2\,c_2}$. Per ser més exactes, pretenia demostrar en primera instància aquest enunciat i, de corol·lari, treure'n la caracterització de $\mathbf{a=b=c=\frac{k}{3}}$ com a cas límit per al qual el producte $\mathbf{a\,b\,c}$ assoleix el màxim: si $\mathbf{c-a=0}$ (circumstància que només s'acomplirà si $\mathbf{a=b=c}$), $\mathbf{a\,b\,c}$ seria més gran que en qualsevol altre cas. I va ser en tractar infructuosament de demostrar això, organitzant una tipologia de casos que convenia il·lustrar amb exemples per no fer tan abstracta l'argumentació, quan va aparèixer un contraexemple que ho engegava tot en orris: en el cas $\mathbf{1+5+6=12=2+2+8}$, $\mathbf{6-1=5<6=8-2}$, i malgrat això era $\mathbf{1\times5\times6=30<32=2\times2\times8}$. Tres fulls endavant veureu que l'excepció ja es dóna amb $\mathbf{k=11}$ ($\mathbf{1+5+5=11=2+2+7}$) i que, a mesura que augmentem $\mathbf{k}$, n'augmenta el nombre, fins al punt que és pertinent preguntar-se si aquests casos són realment una "excepció" o, més enllà de $\mathbf{k=21}$ en què ens hem aturat, el que fins la descoberta esmentada crèiem universal i que ara podríem caure en el parany de qualificar de "norma" per salvar la cara, no serà en realitat una excepció tant més rara, estadísticament parlant, quant més elevats siguin els valors $\mathbf{k}$ tractats.

Però, tot i haver rectificat l'orientació del discurs, moure's en el domini dels nombres enters resultava feixuc i, com que tampoc estava disposat a esmerçar-hi molt de temps, va veure que tot resultava més senzill passant al dels reals, tot considerant que en el subdomini positiu el producte $\mathbf{a\,b\,c}$ era una funció contínua. Limitar-se després a treballar amb enters ja era una opció pròpia de l'aplicació, així que l'autor va optar per esbrinar en quines condicions $\mathbf{a\,b\,c}$ esdevenia màxim, però no pas com una funció de tres variables $\mathbf{a}$, $\mathbf{b}$ i $\mathbf{c}$, atès que amb $\mathbf{k}$ constant la tercera quedava determinada pel valor de les altres dues, sinó de dues variables.

Si escollim $\mathbf{a}$ i $\mathbf{c}$ com les variables independents (amb $\mathbf{b=k-a-c}$) i definim la funció producte $\mathbf{f(a,c)=a\,b\,c=a\,c\,(k-a-c)=a\,c\,k-a^2c-a\,c^2}$, mirarem si aquesta té algun màxim local, que (com a condició necessària) correspondria als valors per als quals s'anul·len les primeres derivades parcials

$$\mathbf{f'_a=c\,k-2\,c\,a-c^2=0}$$

$$\mathbf{f'_c=a\,k-a^2-2\,a\,c=0}$$

Com que $\mathbf{a\ne0}$ i $\mathbf{c\ne0}$, podem dividir els dos membres de la primera equació per $\mathbf{c}$ i els de la segona per $\mathbf{a}$, donant lloc a les dues equacions lineals

$$\mathbf{k-2\,a-c=0}$$

$$\mathbf{k-a-2\,c=0}$$ de fàcil resolució: $\mathbf{c=k-2\,a}$ i $\mathbf{k-a-2\,(k-2\,a)=k+3\,a=0}$, d'on tenim $\mathbf{a=\frac{k}{3}}$ i $\mathbf{c=k-2\,\frac{k}{3}=\frac{3\,k-2\,k}{3}=\frac{k}{3}}$.

El valor $\mathbf{b}$ corresponent als d'$\mathbf{a}$ i $\mathbf{c}$ és $\mathbf{b=k-a-c=k-\frac{k}{3}-\frac{k}{3}=\frac{3\,k-2\,k}{3}=\frac{k}{3}}$.

Per saber si la tríada crítica $\mathbf{a=\frac{k}{3}}$, $\mathbf{b=\frac{k}{3}}$ i $\mathbf{c=\frac{k}{3}}$ representa efectivament un màxim haurem de calcular les quatre segones derivades parcials i formar amb elles l'anomenada matriu Hessiana, mirant si el determinant és positiu i si els elements principals són negatius (condicions suficients). Les quatre derivades parcials són

$f''_{aa} = -2c = -\frac{2}{3}k$

$f''_{cc} = -2a = -\frac{2}{3}k$

$f''_{ac} = f''_{ca} = k - 2a - 2c = k - \frac{2}{3}k - \frac{2}{3}k = \frac{3k - 4k}{3} = -\frac{k}{3}$ i, a partir d'això, veiem que $f''_{aa} f''_{cc} - f''_{ac} f''_{ca} = \frac{4}{9}k^2 - \frac{1}{9}k^2 = \frac{3}{9}k^2 = \frac{k^2}{3} > 0$ i que $f''_{aa} = f''_{cc} < 0$, fet que confirma el caràcter de màxim local del punt $\frac{k}{3}, \frac{k}{3}, \frac{k}{3}$ corresponent al producte $\left(\frac{k}{3}\right)^3 = \frac{k^3}{27}$, únic en el domini restringit de la funció $\frac{k}{3} \geq a > 0,\ b > 0,\ c \geq \frac{k}{3} > 0$ i en el més general $a > 0,\ b > 0,\ c > 0$.

Hem volgut oferir aquesta demostració per motivacions més estètiques que ètiques, atès que l'evidència que el producte màxim corresponia a $\frac{k^3}{27}$ (o a l'enter inferior més pròxim, si **k** no era múltiple de **3**) ja la teníem, en haver dissenyat una funció **3+*** pensada per obtenir llistes amb totes les combinacions d'enters **a**, **b** i **c** que satisfan la condició **a + b + c = k** amb l'argument **k** subministrat (**K** en el codi):

```
(defun 3+* (K / A B C L)
  (if (or (/= (type K) 'INT) (< K 3))
      (progn
        (prompt "\nError: l'argument ha de ser un enter > 2.")
        (princ))
      (progn
        (setq C (1- K))
        (while (>= (setq C (1- C)) (/ K 3.0))
               (setq A 0 B (- K C))
               (while (<= (setq A (1+ A)) (setq B (1- B)))
                      (if (<= B C)
                          (setq L (cons (list (list (* A B C)) A B C) L))))))))
```

Cada element de **L** és una llista composta pels valors **a**, **b** i **c**, precedits per una sublista amb el producte **a b c**. L'element inicial correspondrà al producte més elevat, amb $a = b = c = \frac{k}{3}$ (quan $\frac{k}{3}$ sigui exacte), $a = b = c - 1$ (**c** = part entera de $\frac{k}{3}$) o $a = b - 1$ (**b = c =** part entera de $\frac{k}{3}$), i l'últim al producte més baix, amb $a = b = 1$ i $c = k - 2$. Perquè el lector se'n faci una idea i pugui veure les distribucions possibles (entre 3 requadres 9×3) sobre un total de **k** emplenaments, reproduïm els resultats corresponents als arguments $3 \leq k \leq 21$ (els primers valors $3 \leq k \leq 9$ poden servir per a la distribució d'emplenaments d'un requadre 9×3 entre les seves 3 línies):

```
(3+* 3):  (((1) 1 1 1))

(3+* 4):  (((2) 1 1 2))

(3+* 5):  (((4) 1 2 2) ((3) 1 1 3))

(3+* 6):  (((8) 2 2 2) ((6) 1 2 3) ((4) 1 1 4))

(3+* 7):  (((12) 2 2 3) ((9) 1 3 3) ((8) 1 2 4) ((5) 1 1 5))

(3+* 8):  (((18) 2 3 3) ((16) 2 2 4) ((12) 1 3 4) ((10) 1 2 5) ((6) 1 1 6))

(3+* 9):  (((27) 3 3 3) ((24) 2 3 4) ((16) 1 4 4) ((20) 2 2 5) ((15) 1 3 5)
           ((12) 1 2 6) ((7) 1 1 7))

(3+* 10): (((36) 3 3 4) ((32) 2 4 4) ((30) 2 3 5) ((20) 1 4 5) ((24) 2 2 6)
           ((18) 1 3 6) ((14) 1 2 7) ((8) 1 1 8))

(3+* 11): (((48) 3 4 4) ((45) 3 3 5) ((40) 2 4 5) ((25) 1 5 5) ((36) 2 3 6)
           ((24) 1 4 6) ((28) 2 2 7) ((21) 1 3 7) ((16) 1 2 8) ((9) 1 1 9))
```

```
(3+* 12): (((64) 4 4 4) ((60) 3 4 5) ((50) 2 5 5) ((54) 3 3 6) ((48) 2 4 6)
           ((30) 1 5 6) ((42) 2 3 7) ((28) 1 4 7) ((32) 2 2 8) ((24) 1 3 8)
           ((18) 1 2 9) ((10) 1 1 10))

(3+* 13): (((80) 4 4 5) ((75) 3 5 5) ((72) 3 4 6) ((60) 2 5 6) ((36) 1 6 6)
           ((63) 3 3 7) ((56) 2 4 7) ((35) 1 5 7) ((48) 2 3 8) ((32) 1 4 8)
           ((36) 2 2 9) ((27) 1 3 9) ((20) 1 2 10) ((11) 1 1 11))

(3+* 14): (((100) 4 5 5) ((96) 4 4 6) ((90) 3 5 6) ((72) 2 6 6) ((84) 3 4 7)
           ((70) 2 5 7) ((42) 1 6 7) ((72) 3 3 8) ((64) 2 4 8) ((40) 1 5 8)
           ((54) 2 3 9) ((36) 1 4 9) ((40) 2 2 10) ((30) 1 3 10) ((22) 1 2 11)
           ((12) 1 1 12))

(3+* 15): (((125) 5 5 5) ((120) 4 5 6) ((108) 3 6 6) ((112) 4 4 7) ((105) 3 5 7)
           ((84) 2 6 7) ((49) 1 7 7) ((96) 3 4 8) ((80) 2 5 8) ((48) 1 6 8)
           ((81) 3 3 9) ((72) 2 4 9) ((45) 1 5 9) ((60) 2 3 10) ((40) 1 4 10)
           ((44) 2 2 11) ((33) 1 3 11) ((24) 1 2 12) ((13) 1 1 13))

(3+* 16): (((150) 5 5 6) ((144) 4 6 6) ((140) 4 5 7) ((126) 3 6 7) ((98) 2 7 7)
           ((128) 4 4 8) ((120) 3 5 8) ((96) 2 6 8) ((56) 1 7 8) ((108) 3 4 9)
           ((90) 2 5 9) ((54) 1 6 9) ((90) 3 3 10) ((80) 2 4 10) ((50) 1 5 10)
           ((66) 2 3 11) ((44) 1 4 11) ((48) 2 2 12) ((36) 1 3 12) ((26) 1 2 13)
           ((14) 1 1 14))

(3+* 17): (((180) 5 6 6) ((175) 5 5 7) ((168) 4 6 7) ((147) 3 7 7) ((160) 4 5 8)
           ((144) 3 6 8) ((112) 2 7 8) ((64) 1 8 8) ((144) 4 4 9) ((135) 3 5 9)
           ((108) 2 6 9) ((63) 1 7 9) ((120) 3 4 10) ((100) 2 5 10) ((60) 1 6 10)
           ((99) 3 3 11) ((88) 2 4 11) ((55) 1 5 11) ((72) 2 3 12) ((48) 1 4 12)
           ((52) 2 2 13) ((39) 1 3 13) ((28) 1 2 14) ((15) 1 1 15))

(3+* 18): (((216) 6 6 6) ((210) 5 6 7) ((196) 4 7 7) ((200) 5 5 8) ((192) 4 6 8)
           ((168) 3 7 8) ((128) 2 8 8) ((180) 4 5 9) ((162) 3 6 9) ((126) 2 7 9)
           ((72) 1 8 9) ((160) 4 4 10) ((150) 3 5 10) ((120) 2 6 10) ((70) 1 7 10)
           ((132) 3 4 11) ((110) 2 5 11) ((66) 1 6 11) ((108) 3 3 12)
           ((96) 2 4 12) ((60) 1 5 12) ((78) 2 3 13) ((52) 1 4 13) ((56) 2 2 14)
           ((42) 1 3 14) ((30) 1 2 15) ((16) 1 1 16))

(3+* 19): (((252) 6 6 7) ((245) 5 7 7) ((240) 5 6 8) ((224) 4 7 8) ((192) 3 8 8)
           ((225) 5 5 9) ((216) 4 6 9) ((189) 3 7 9) ((144) 2 8 9) ((81) 1 9 9)
           ((200) 4 5 10) ((180) 3 6 10) ((140) 2 7 10) ((80) 1 8 10)
           ((176) 4 4 11) ((165) 3 5 11) ((132) 2 6 11) ((77) 1 7 11)
           ((144) 3 4 12) ((120) 2 5 12) ((72) 1 6 12) ((117) 3 3 13)
           ((104) 2 4 13) ((65) 1 5 13) ((84) 2 3 14) ((56) 1 4 14) ((60) 2 2 15)
           ((45) 1 3 15) ((32) 1 2 16) ((17) 1 1 17))

(3+* 20): (((294) 6 7 7) ((288) 6 6 8) ((280) 5 7 8) ((256) 4 8 8) ((270) 5 6 9)
           ((252) 4 7 9) ((216) 3 8 9) ((162) 2 9 9) ((250) 5 5 10) ((240) 4 6 10)
           ((210) 3 7 10) ((160) 2 8 10) ((90) 1 9 10) ((220) 4 5 11)
           ((198) 3 6 11) ((154) 2 7 11) ((88) 1 8 11) ((192) 4 4 12)
           ((180) 3 5 12) ((144) 2 6 12) ((84) 1 7 12) ((156) 3 4 13)
           ((130) 2 5 13) ((78) 1 6 13) ((126) 3 3 14) ((112) 2 4 14)
           ((70) 1 5 14) ((90) 2 3 15) ((60) 1 4 15) ((64) 2 2 16) ((48) 1 3 16)
           ((34) 1 2 17) ((18) 1 1 18))

(3+* 21): (((343) 7 7 7) ((336) 6 7 8) ((320) 5 8 8) ((324) 6 6 9) ((315) 5 7 9)
           ((288) 4 8 9) ((243) 3 9 9) ((300) 5 6 10) ((280) 4 7 10)
           ((240) 3 8 10) ((180) 2 9 10) ((100) 1 10 10) ((275) 5 5 11)
           ((264) 4 6 11) ((231) 3 7 11) ((176) 2 8 11)
           ((99) 1 9 11) ((240) 4 5 12) ((216) 3 6 12) ((168) 2 7 12)
           ((96) 1 8 12) ((208) 4 4 13) ((195) 3 5 13) ((156) 2 6 13)
           ((91) 1 7 13) ((168) 3 4 14) ((140) 2 5 14) ((84) 1 6 14)
           ((135) 3 3 15) ((120) 2 4 15) ((75) 1 5 15) ((96) 2 3 16)
           ((64) 1 4 16) ((68) 2 2 17) ((51) 1 3 17) ((36) 1 2 18) ((19) 1 1 19))
```

I ja que abans us hem parlat de la hipòtesis $\mathbf{c_1 - a_1 < c_2 - a_2 \Rightarrow a_1\,b_1\,c_1 > a_2\,b_2\,c_2}$ que finalment s'havia revelat falsa, amb excepcions que creixien amb **k**, tanquem el capítol amb la funció **EXC** que ens en proporciona llistes (també per a $\mathbf{3 \le k \le 21}$):

```
(defun EXC (L / CAARE CADRE LASTE LASTE-CADRE LEXC)
  (foreach E1 L
    (setq CAARE (caar E1) CADRE (cadr E1) LASTE (last E1)
          LASTE-CADRE (- LASTE CADRE) L (cdr L))
    (if L (foreach E2 L
            (if (and (<= CAARE (caar E2)) (< LASTE-CADRE (- (last E2) (cadr E2))))
                (setq LEXC (cons (list E1 E2) LEXC))))))
  (reverse LEXC))

(EXC (3+* 11)): ((((25) 1 5 5) ((28) 2 2 7)))

(EXC (3+* 12)): ((((30) 1 5 6) ((32) 2 2 8)))

(EXC (3+* 13)): ((((36) 1 6 6) ((48) 2 3 8))
                 (((36) 1 6 6) ((36) 2 2 9))
                 (((35) 1 5 7) ((36) 2 2 9)))

(EXC (3+* 14)): ((((72) 2 6 6) ((72) 3 3 8))
                 (((42) 1 6 7) ((54) 2 3 9))
                 (((40) 1 5 8) ((40) 2 2 10)))

(EXC (3+* 15)): ((((49) 1 7 7) ((72) 2 4 9))
                 (((49) 1 7 7) ((60) 2 3 10))
                 (((48) 1 6 8) ((60) 2 3 10)))

(EXC (3+* 16)): ((((98) 2 7 7) ((108) 3 4 9))
                 (((56) 1 7 8) ((80) 2 4 10))
                 (((56) 1 7 8) ((66) 2 3 11))
                 (((54) 1 6 9) ((66) 2 3 11)))

(EXC (3+* 17)): ((((112) 2 7 8) ((120) 3 4 10))
                 (((64) 1 8 8) ((100) 2 5 10))
                 (((64) 1 8 8) ((99) 3 3 11))
                 (((64) 1 8 8) ((88) 2 4 11))
                 (((64) 1 8 8) ((72) 2 3 12))
                 (((63) 1 7 9) ((88) 2 4 11))
                 (((63) 1 7 9) ((72) 2 3 12))
                 (((60) 1 6 10) ((72) 2 3 12)))

(EXC (3+* 18)): ((((128) 2 8 8) ((150) 3 5 10))
                 (((128) 2 8 8) ((132) 3 4 11))
                 (((126) 2 7 9) ((132) 3 4 11))
                 (((72) 1 8 9) ((110) 2 5 11))
                 (((72) 1 8 9) ((108) 3 3 12))
                 (((72) 1 8 9) ((96) 2 4 12))
                 (((72) 1 8 9) ((78) 2 3 13))
                 (((70) 1 7 10) ((96) 2 4 12))
                 (((70) 1 7 10) ((78) 2 3 13))
                 (((66) 1 6 11) ((78) 2 3 13)))

(EXC (3+* 19)): ((((192) 3 8 8) ((200) 4 5 10))
                 (((144) 2 8 9) ((165) 3 5 11))
                 (((144) 2 8 9) ((144) 3 4 12))
                 (((81) 1 9 9) ((132) 2 6 11))
                 (((81) 1 9 9) ((144) 3 4 12))
                 (((81) 1 9 9) ((120) 2 5 12))
                 (((81) 1 9 9) ((117) 3 3 13))
                 (((81) 1 9 9) ((104) 2 4 13))
                 (((81) 1 9 9) ((84) 2 3 14))
                 (((140) 2 7 10) ((144) 3 4 12))
                 (((80) 1 8 10) ((120) 2 5 12))
                 (((80) 1 8 10) ((117) 3 3 13))
                 (((80) 1 8 10) ((104) 2 4 13))
                 (((80) 1 8 10) ((84) 2 3 14))
                 (((77) 1 7 11) ((104) 2 4 13))
                 (((77) 1 7 11) ((84) 2 3 14))
                 (((72) 1 6 12) ((84) 2 3 14)))
```

```
(EXC (3+* 20)): ((((216) 3 8 9) ((220) 4 5 11))
                 (((162) 2 9 9) ((198) 3 6 11))
                 (((162) 2 9 9) ((192) 4 4 12))
                 (((162) 2 9 9) ((180) 3 5 12))
                 (((160) 2 8 10) ((180) 3 5 12))
                 (((90) 1 9 10) ((144) 2 6 12))
                 (((90) 1 9 10) ((156) 3 4 13))
                 (((90) 1 9 10) ((130) 2 5 13))
                 (((90) 1 9 10) ((126) 3 3 14))
                 (((90) 1 9 10) ((112) 2 4 14))
                 (((90) 1 9 10) ((90) 2 3 15))
                 (((154) 2 7 11) ((156) 3 4 13))
                 (((88) 1 8 11) ((130) 2 5 13))
                 (((88) 1 8 11) ((126) 3 3 14))
                 (((88) 1 8 11) ((112) 2 4 14))
                 (((88) 1 8 11) ((90) 2 3 15))
                 (((84) 1 7 12) ((112) 2 4 14))
                 (((84) 1 7 12) ((90) 2 3 15))
                 (((78) 1 6 13) ((90) 2 3 15)))

(EXC (3+* 21)): ((((243) 3 9 9) ((264) 4 6 11))
                 (((240) 3 8 10) ((240) 4 5 12))
                 (((180) 2 9 10) ((216) 3 6 12))
                 (((180) 2 9 10) ((208) 4 4 13))
                 (((180) 2 9 10) ((195) 3 5 13))
                 (((100) 1 10 10) ((168) 2 7 12))
                 (((100) 1 10 10) ((195) 3 5 13))
                 (((100) 1 10 10) ((156) 2 6 13))
                 (((100) 1 10 10) ((168) 3 4 14))
                 (((100) 1 10 10) ((140) 2 5 14))
                 (((100) 1 10 10) ((135) 3 3 15))
                 (((100) 1 10 10) ((120) 2 4 15))
                 (((176) 2 8 11) ((195) 3 5 13))
                 (((99) 1 9 11) ((156) 2 6 13))
                 (((99) 1 9 11) ((168) 3 4 14))
                 (((99) 1 9 11) ((140) 2 5 14))
                 (((99) 1 9 11) ((135) 3 3 15))
                 (((99) 1 9 11) ((120) 2 4 15))
                 (((168) 2 7 12) ((168) 3 4 14))
                 (((96) 1 8 12) ((140) 2 5 14))
                 (((96) 1 8 12) ((135) 3 3 15))
                 (((96) 1 8 12) ((120) 2 4 15))
                 (((96) 1 8 12) ((96) 2 3 16))
                 (((91) 1 7 13) ((120) 2 4 15))
                 (((91) 1 7 13) ((96) 2 3 16))
                 (((84) 1 6 14) ((96) 2 3 16)))
```

Convé aclarir que la intuició manifestada quatre fulls enrera no s'ha confirmat. Tot i no haver pogut arribar a una conclusió exacta, per limitacions tècniques, sí que hem trobat prou elements per llançar una hipòtesi qualitativament plausible sobre l'evolució del nombre de casos que anomenàvem "excepcionals" en relació als que qualificàvem de "normals". Quantificar aquests casos ha estat tan senzill com crear una funció **NORM**, calcada d'**EXC** llevat en la línia de codi

(and (<= CAARE (caar E2)) (< LASTE-CADRE (- (last E2) (cadr E2)))

que cambiava a

(and (> CAARE (caar E2)) (< LASTE-CADRE (- (last E2) (cadr E2))).

Sols ha calgut fer **(length (EXC (3+* <k>)))** i **(length (NORM (3+* <k>)))** per copsar que la primera xifra augmentava amb **k** però, com que la segona també ho feia, per veure en què quedava tot plegat ha estat indefugible comparar els dos creixements. El quocient entre els valors homòlegs d'ambdues sèries semblava un bon indicador, així que ens hi hem posat i, malgrat que el recull de dades ha estat força limitat (més enllà de de **k** = **210** l'ordinador es penjava), ja n'hi hagut prou per desterrar la hipòtesi de rocambolescos girs de truita en considerar valors més elevats de **k**. En conclusió, no sols no s'acomplirà la sospita abans esmentada (que hi hauria un valor **k** a partir del qual els que crèiem casos "excepcionals" superarien els que crèiem "normals", amb un quocient més gran que **1**) sinó tampoc la que en seria una versió atenuada (que els dos valors convergirien a mesura que augmentés **k**, amb un

quocient que tendiria a **1**): si dibuixeu la gràfica cartesiana de **k** i el quocient (que convindrà representar a una escala d'ampliació 100) veureu que aquest tendeix a estabilitzar-se asimptòticament a un valor probablement situat per sota del **12%**, i això faria bona la classificació inicial en casos normals i excepcionals, tot i que potser sería més encertat parlar de casos més o menys freqüents (o majoritaris i minoritaris) atès que, a partir de **k** = **11**, els segons deixen de ser una excepció perquè apareixen sempre. Acabem amb el recull de les mostres efectuades (limitades als valors **k** múltiples de **3**, per evitar oscil·lacions que podrien distreure), que a patir de l'intèrval **11 = k ≤ 21** hem espaiat, practicant-les de **21** en **21** unitats: **1/62 = 0,0161 (12); 3/158 = 0,0190 (15); 10/320 = 0,0313 (18); 26/596 = 0,0436 (21); 669/9.264 = 0,0722 (42); 3.947/46.418 = 0,0850 (63); 13.235/145.693 = 0,0908 (84); 33.698/354.346 = 0,0951 (105); 71.474/732.574 = 0,0976 (126); 135.020/1.354.552 = 0,0997 (147); 233.014/2.307.041 = 0,1010 (168); 377.340/3.691.220 = 0,1022 (189); 579.095/5.620.315 = 0,1030 (210).** En definitiva, no només és veritat que el màxim producte es produeix per a valors $\mathbf{a = b = c = \frac{k}{3}}$; estadísticament podem afirmar que és altament probable que, quan $\mathbf{c_1 - a_1 < c_2 - a_2}$, sigui $\mathbf{a_1\, b_1\, c_1 > a_2\, b_2\, c_2}$, cosa encara més palesa si, en lloc de la ràtio utilitzada (casos_excepcionals/casos_normals), considerem casos_excepcionals/total_de_casos: **1/63 = 0,0159 (12); 3/161 = 0,0186 (15); 10/330 = 0,0303 (18); 26/622 = 0,0418 (21); 669/9.933 = 0,0674 (42); 3.947/50.365 = 0,0784 (63); 13.235/158.928 = 0,0833 (84); 33.698/388.044 = 0,0868 (105); 71.474/804.048 = 0,0889 (126); 135.020/1.489.572 = 0,0906 (147); 233.014/2.540.055 = 0,0917 (168); 377.340/4.068.560 = 0,0927 (189); 579.095/6.199.410 = 0,0934 (210).**

Epíleg: això acaba igual que el conte de la sopa de còdols?

L'autor recordava d'un llibre de lectures de tercer o quart curs del batxillerat que va cursar (el de Joaquín Ruiz-Giménez, previ al de la *Ley General de Educación* de 1970), que tindrà en algun racó de casa seva però no ha sabut trobar, un conte anònim de tall picaresc que després ha localitzat a Google, en versions diferents (un soldat, un rodamón, un pelegrí o un frare com a protagonistes) de procedència també diversa (Castella, Galícia i Portugal, país on a la ciutat d'Almeirim hi ha una especialitat gastronòmica anomenada *sopa de pedra*) i que tractarà de resumir: un personatge itinerant demana aixopluc en una casa de pagès, la mestressa accepta que passi la nit en un racó però no sembla gens amatent a donar-li res de menjar; aleshores el personatge comenta que no tindrà més remei que utilitzar un recurs de supervivència que molt pocs coneixen i que cuinarà una sopa de còdols si li posen una olla al foc, amb aigua i una mica de sal, mentre ell surt un moment a collir uns quants còdols en una riera pròxima; no cal dir que la mestressa, encuriosida, hi accedeix; quan el coneixedor d'una recepta tan pràctica com econòmica torna a la casa i aboca els còdols a l'aigua bullent, com qui no vol la cosa li diu a la mestressa que la preparació ja està en marxa però que el resultat pot millorar si s'hi afegeix una mica d'oli i unes patates, ingredients que aquesta li porta sense pensar-s'ho dos cops; quan tot és a l'olla, el singular cuiner se'n recorda que un grapat de mongetes també hi ajudarien (arribats aquí, cada versió farceix el relat amb la inclusió de menges locals com cansalada, xoriço, llonza, etc. que, malgrat una suspicàcia creixent, la dona li va subministrant sense gosar replicar-li res); finalment, abans d'entaular-se i posar-se a menjar la sopa, l'hoste treu de l'olla els còdols, tot aclarint-li a l'astorada amfitriona que els pot llençar, perquè ja han deixat anar tota la substància; la mestressa s'adona que li han pres el pèl i, avergonyida, fa mutis. És enginyós, però us preguntareu a què ve ara aquest relat.

Doncs digueu solucions normalitzades en comptes de còdols i ja teniu la resposta, tret d'una premeditació prou evident en el cas del protagonista del conte però que l'autor us pot ben assegurar que era absent de les seves intencions. Fora d'això, el paral·lelisme és total:

- La idea de limitar-se a solucions normalitzades ja era present en el capítol 1 (*Inventariar les solucions: un cul-de-sac*). En la seva beneiteria, l'autor creia que amb aquesta argúcia la formació de l'inventari de solucions esdevindria més assequible (en passar de **6.670.903.752.021.072.936.960** SUDOKUs-SOLUCIÓ a "només" **5.147.302.277.794.037.760**, tot i que encara ignorés la primera xifra i únicament pensés a reduir una quantitat molt alta (però que no sospitava que pogués ser-ho tant com després de llegir la referència als treballs de Felgenhauer i Jarvis) a la seva 1.296-èsima part.
- Tot i haver deixat enrera aquest camí i trobar-nos en un context ben diferent, en el capítol 10 (*Solucions compatibles ho són totes les que hi són...*) reprenem la idea de l'estalvi que pot reportar la utilització de solucions normalitzades, en permetre'ns de restringir el nombre de comparacions a les funcions **PERFIL-V*V** i **COMPAT-PERFILS**.
- Però, gairebé sense adonar-nos, anem alimentant expectatives exagerades respecte a les virtuts de la substitució dels subconjunts de 1.296 SUDOKUs-SOLUCIÓ per un d'ells (el normalitzat), oblidant que cerquem tota mena de solucions compatibles amb la part pública d'una solució normalitzada (aquelles caselles que volem que formin part d'un SUDOKU-PROBLEMA capaç de determinar la variant normalitzada del SUDOKU-SOLUCIÓ) i que moltes d'aquestes no ho seran, de normalitzades. És en el capítol 11 (*... però no hi eren totes les que ho són*) quan situem els beneficis de l'artifici en la seva justa mesura, afegint als avantatges de reduir l'àmbit d'actuació de **PERFIL-V*V** i **COMPAT-PERFILS** l'encert de subdividir l'inventari de solucions compatibles en dues etapes: cerca limitada a les solucions compatibles normalitzades (amb la utilització puntual d'aquestes dues funcions, seguida de les actualitzacions rutinàries a càrrec de **NUMCOMPAT**), i cerca generalitzada de tota mena de solucions compatibles (amb la utilització puntual d'**ANORMALS** i les actualitzacions rutinàries a càrrec de **NUMCOMPAT**) quan de normalitzades només en queda una.
- En aquesta situació arribem a preguntar-nos si no sortiria més a compte unificar el procés d'inventari, prescindint de l'artifici de les solucions normalitzades i de considerar el moment **FOTO2** (una sola solució normalitzada compatible) com a transició de la 1ª a la 2ª etapa, confiant-ho exclusivament tot a la intervenció

d'**ANORMALS** i de **NUMCOMPAT**. Però afirmem que s'han fet proves i que la subdivisió sí que surt a compte. N'hi hauria prou a llegir, en la pàgina 577 del capítol 12 (*Un nou camí i també una drecera*), el paràgraf següent:

> "Si la decisió de fragmentar el procés de recompte de solucions en les tres etapes "PRE-FOTO", "FOTO 1-2" i "FOTO 2-2" del capítol precedent, desdoblant els elements separadors en dos, **(PERFIL-V*V)** + **(COMPAT-PERFILS)** d'una banda i **(ANORMALS)** de l'altra, representatius d'estratègies de resolució ben diferents (en comptes de deixar-se de "solucions normalitzades" i d'altres històries, i limitar-se a dues etapes "PRE-FOTO" i "POST-FOTO" separades per un algorisme del tipus **(ANORMALS)**), va ser el típic pegat per falcar una construcció que quedava coixa per una badada inicial, però que unes succintes comprovacions van mostrar que bàsicament havia estat encertada (una vegada més, allò de "*sonó la flauta por casualidad*"), el dilema que se'ns obre ara, en suggerir un mètode alternatiu, és més compromés. Perquè el problema és saber què surt més a compte, a partir del moment en què **SUD-MIN** activarà **POST**: muntar guàrdia davant de l'ordinador, amatents als relativament soportables temps d'espera que s'aniran succeint, cada cop més breus, fins a la conclusió; o assumir la perspectiva de suportar dos períodes d'espera més llargs, quan **COMPAT-PERFILS** faci inventari de les solucions normalitzades compatibles i **ANORMALS** faci el mateix, d'una altra forma, amb les simplement compatibles, amb la possibilitat d'aprofitar les pauses per anar a dinar, a sopar... o a dormir."

La segona frase fa referència a l'eficiència relativa dels mètodes 1 i 2, que a continuació l'autor resol diplomàticament amb un veredicte d'empat tècnic (que els successius apèndixs al capítol, relatius a l'adaptació del conjunt del codi a les millores introduïdes en l'opció "A" d'obtenció del SUDOKU-SOLUCIÓ en els capítols 5, 6 i 7, que repercuteixen en el mètode 1, s'encarregaran de corregir a favor de la seva superioritat), però la primera al·ludeix inequívocament a la possibilitat d'estar-se de solucions normalitzades i confiar la cerca des de bon principi a la funció **ANORMALS** i, tot i així, es ratifica en l'encert d'haver-les utilitzat, sigui o no xiripa.

- Tanmateix el capítol precedent havia acabat amb una constatació inesperada, comú als nous mètodes 3 i 4: gràcies a **ANORMALS** el SUDOKU-PROBLEMA quedava determinat sense esperes significatives. Llavors, ¿en què quedem? ¿La utilització exclusiva d'**ANORMALS** funcionava o no? ¿Que potser les esmentades "succintes comprovacions" no s'havíen fet i tot plegat era un muntatge de l'autor per amagar un nou fracàs en l'intent de justificar el recurs a solucions normalitzades? L'autor proclama solemnement que les comparacions es van abordar i que a totes elles, utilitzant solucions i seqüències d'emplenament de caselles en què l'aplicació del mètode 2 donava lloc a temps d'espera suportables, en modificar **FES-SUDOKU-2** de forma que el tàndem **PERFIL-V*V** + **COMPAT-PERFILS** quedés suplantat per **ANORMALS**, substituint l'expressió

```
(progn
  (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
  (RETOL-2 "Determinar" "quantes"  "solucions" "normalitza-" "des són"
           "compatibles" "pot trigar" "hores." "Espereu...")
  (PERFIL-V*V)
  (COMPAT-PERFILS)
  (command "DESHACER" "R" "UY" "ESPACIOM"))
```

per

```
(progn
  (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
  (RETOL-2 "Determinar" "quantes"  "solucions" "ordinàries" "són"
           "compatibles" "pot trigar" "hores." "Espereu...")
  (ANORMALS ())
  (prompt (strcat "\n.\nAmb " (itoa KK) " solucions heu d'omplir més "
                  "caselles.\n"))
  (command "DESHACER" "R" "UY" "ESPACIOM"))
```

l'espera s'eternitzava. Fins al punt que, per dissipar la sospita d'haver comès algun error de programació que hagués dut a un bucle sense fi, l'autor va posar a l'expressió inicial de **RE+MEMBER**,

```
(if (COMPLET-9*9 R-9*9)
    (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9) SUDOKUS-V*V)
          KKK (cons T KKK)) ...)
```

un dispositiu que l'informés en temps real del nombre de solucions compatibles detectades

```
(if (COMPLET-9*9 R-9*9)
    (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9) SUDOKUS-V*V)
              KKK (cons T KKK) W (princ " ") W (princ (length KKK)) ...)
```

i el resultat va ser concloent: el procés rutllava força bé, però l'apilament d'arguments i variables locals propi de la dinàmica recursiva de l'exploració, així com el creixement de la llista **SUDOKUS-V*V** era la causa que una velocitat inicialment acceptable anés minvant, fins arribar al col·lapse total amb poc més de vint mil solucions compatibles detectades. Això quan hi havia més de vint mil solucions ordinàries compatibles amb un emplenament que, al seu pas per **SUD-MIN**, provocava l'activació de **POST**, com és el cas del sudoku difícil de LA VANGUARDIA del 25-02-08, de què ens hem ocupat al llarg dels capítols 9 a 12 i que havíem aconseguit que en 3 minuts i 10 segons ens informés que entre aquestes solucions n'hi havia 1.117 de normalitzades. Per contra, amb la substitució esmentada el procés quedava ofegat per manca de memòria: començava amb unes 1.800 solucions detectades per minut, velocitat que anava disminuint i que, quan ja se situava en unes 1.200 solucions per minut, al cap de 14 minuts, l'ordinador es penjava en haver arribat a trobar-ne 21.648. Tot i que tampoc no podem perdre de vista:

- Que aquests resultats no incorporaven la millora just introduïda en el capítol precedent, consistent a treballar amb el sudoku transposat quan en l'original l'ocupació dels requadres 9×3 és més heterogènia. De tota manera, com que la millora, quan era procedent la transposició, afectava en exclusiva la tasca de **PERFIL-V*V** + **COMPAT-PERFILS** (**ANORMALS** en quedava exclosa), més a favor nostre.
- Que les comprovacions s'havien limitat a SUDOKUs-SOLUCIÓ de 23 o més caselles plenes, malgrat que ja sabíem que n'hi havia de 17. I, com veurem de seguida, amb SUDOKUs-SOLUCIÓ més reduïdes (més propers a l'activació de **POST**) **ANORMALS** ja li pren avantatge al tàndem **PERFIL-V*V** + **COMPAT-PERFILS**.

Però, fins i tot tenint present aquesta segona consideració, quan en el capítol precedent avaluàvem l'eficàcia dels mètodes 3 o 4, les respostes en desactivar caselles (incloent-hi l'última, en passar-nos de la ratlla deixant-ne només 22 de plenes) eren pràcticament instantànies.

- I és que la contradicció només és aparent: una cosa es aplicar **ANORMALS** com es fa des del mètode 2 original (activant una casella quan el grau d'emplenament és mitjà o desactivant-la quan el grau d'emplenament és elevat) o manipulat tal com mostravèm a la pàgina precedent (activant una casella quan el grau d'emplenament és baix), circumstàncies en què la funció ha de localitzar totes les solucions compatibles (la cerca pot ser curta en els dos primers casos, però en el tercer de segur que es dispara), i una altra de ben diferent és fer-ho des del mètode 3 o 4, amb la versió d'**ANORMALS** presentada al final del capítol precedent (en què l'execució se subordinava a la condició **(if (or 1-2<34 (< (length KKK) 2)) ...)**) i l'argument **1-2<34** era **nil**. Si en aquest últim cas (desactivant una casella) la cerca quasi sempre és més ràpida que usant el mètode 2 amb un grau d'emplenament similar (si l'única solució compatible és normalitzada, serà al revés a causa de **NORM=1**), la raó és ben senzilla: amb l'estratègia de caminar a recules des d'un estat que només disposa d'una solució compatible, sols cal saber si encara tenim una única solució compatible (i, amb un grau d'emplenament elevat o mitjà, això és arribar i moldre) o ja n'hi ha més (si en trobem dues no caldrà buscar més i, amb graus d'ocupació mitjans o baixos, això també ho farem en un tres i no res).
- Això ja havia quedat prou clar en el capítol precedent, en introduir en **ANORMALS** el segon argument binari **1-2<34** que, quan és **nil**, interromp les autorecursions de **RE+MEMBER** (a la cerca de més solucions compatibles amb l'emplenament actual) tan bon punt n'ha inventariat dues (**(< (length KKK) 2)**), però semblava que era obligat repetir-ho aquí, davant de la perplexitat que pogués causar el fet que l'acció d'**ANORMALS** en una mateixa situació d'emplenament es traduís en durades tan diferents: tot i que el fotograma fos el mateix, el fet d'arribar-hi passant la pel·lícula cap endavant (activant caselles, a la versió trucada del mètode 2) o cap enrera (desactivant-les, amb els mètodes 3 o 4) condicionava la resposta de **RE+MEMBER**. Per reblar aquesta afirmació, completarem les dades relatives als temps d'execució corresponents al mètode 2 original, al mètode 2 trucat (amb la substitució de **PERFIL-V*V** i **COMPAT-PERFILS** per **ANORMALS**) i als mètode 3 o 4, del sudoku de més amunt (el de LA VANGUARDIA del 25-02-08), de **23 caselles** ocupades, amb les dades corresponents al sudoku que figurava al final de la pàgina 273 del capítol *Etapa prèvia: crear una solució, IV (... o a Camprodon)*, després de les solucions PANÒNICA i CANÒNICA, de **21 caselles** ocupades, i amb les del primer de la col·lecció de 47.793 sudokus bàsics publicats per Gordon Royle (aquesta és la xifra que donàvem en el capítol *Condicions mínimes*, però en escriure això el dia 12/01/2012 ja n'eren 49.151), de **17 caselles** ocupades, i així podrem copsar que:
  - Amb el mètode 2 original, quan abans (amb menys caselles emplenades) s'activi **POST** més trigarà **COMPAT-PERFILS** a fer inventari de les solucions compatibles normalitzades. Activada **POST** amb un mateix nombre de caselles emplenades, quan menys falti (menys caselles s'hagin d'emplenar) per arribar a la solució única menys trigarà **COMPAT-PERFILS** a fer aquest inventari.

  - Amb el mètode 2 trucat, quan abans (amb menys caselles emplenades) s'activi **POST** més trigarà **ANORMALS** a fer inventari de totes les solucions compatibles. Activada **POST** amb un mateix nombre de caselles emplenades, quan menys falti (menys caselles calgui emplenar) per arribar a la solució única menys trigarà **ANORMALS** a fer aquest inventari.
  - Amb els mètodes 3 i 4:
    - A la baixa (havent de desactivar caselles), quant més haguem de recular fins arribar a una solució única no redundant (quantes menys caselles tingui la solució), més llarga serà l'última pausa amb **ANORMALS.**
    - A l'alça (havent d'activar caselles, contingència circumscrita al mètode 3), quantes més caselles haguem de seguir emplenant fins arribar a una solució única, a partir del moment en què en tinguem **NUM** i entri en acció **ANORMALS** (quantes més caselles tingui aquesta solució), més ens farà esperar aquesta.

  En el cas que, utilitzant el mètode 2 trucat, **POST** s'activés precisament en el moment en què l'última casella activada completés un SUDOKU-PROBLEMA, **ANORMALS** treballaria en les mateixes condicions en què ho farien els mètodes 3 o 4 en la penúltima intervenció (no s'entretindria buscant altres solucions compatibles, perquè no n'hi hauria més) i, per tant, la durada d'aquesta seria la mateixa.
- En el sudoku de **21 caselles**, aplicant el métode 2 original **COMPAT-PERFILS** triga 15 segons a fer inventari de les solucions compatibles normalitzades (això si aconseguim que entri en acció amb 19 caselles ocupades, per exemple activant de primer les 7 caselles del requadre 9×3 inferior, després les 8 del superior i finalment els valors **6**, **4**, **9** i **1** del central); amb el métode 2 trucat **ANORMALS** triga 54 segons, i amb el mètode 3 (a la baixa) o 4 totes les intervencions d'**ANORMALS** donen una resposta pràcticament instantània, incloent-hi l'última.
- En el sudoku de **17 caselles**, aplicant el métode 2 original **COMPAT-PERFILS** triga 5 minuts i mig a fer inventari de les solucions compatibles normalitzades; amb el métode 2 trucat **ANORMALS** respon instantàniament, i amb el mètode 3 o 4 totes les intervencions d'**ANORMALS** també donen una resposta pràcticament instantània. La coincidència de resultats pel que fa a **ANORMALS** no té res d'estrany, perquè la 17ª casella completa un SUDOKU-PROBLEMA; tanmateix precisarem que, com que en aquest cas el sudoku és mínim, la identitat de condicions de treball d'**ANORMALS** en el mètode 2 trucat no es produiria amb la penúltima intervenció de la mateixa funció amb els mètodes 3 o 4, sinó amb l'última.
- La conclusió és que, havent proposat un nou mètode (parlem del 3, perquè el 4 és una simple variant i se'n pot prescindir) que, igual com passava amb l'1, és més ràpid que el 2 i reïx en els casos on aquest encalla, podríem desempallegar-nos tranquil·lament de l'últim i amb ell d'un dels elements en què se sustenta: les solucions normalitzades, que han ocupat un espai considerable en aquest treball. Per això sostenim que hi ha una analogia ben clara entre els còdols que el pícar del conte vol presentar a la víctima de l'engany com a ingredient principal de la sopa i les solucions normalitzades que han mantingut cert protagonisme en el discurs que ens ha dut fins aquí: tot i la diferència entre l'innegable caràcter fraudulent dels còdols i la (si més no) respectable intervenció de les solucions normalitzades en el procés heurístic, en ambdós casos queden com a pura anècdota des d'una perspectiva finalista i estem legitimats per excloure'ls del resultat.

Però, igual com en Almeirim és tradició (segons una de les referències obtingudes via Google) que la *sopa de pedra* sigui servida amb un còdol a dins, en arribar el moment de presentar la versió definitiva (pel que fa a l'autor) de SUDÒKULUM, cal prendre una decisió: ¿som expeditius i limitem l'oferta als mètodes 0 (l'anomenada drecera A-0 del diagrama, limitada a solucions obtingudes mitjançant l'opció A), 1 i 3, o som exhaustius i mantenim tant el mètode 2 (amb el "còdol" de les solucions normalitzades) com el 4, per fer bona la dita "més val que en sobri que no que en falti" i perquè sigui el mateix usuari qui en seleccioni uns i refusi els altres?

En cap de les dues opcions ningú no ens eximirà de l'obligació de polir una versió que en el capítol precedent hem deixat operativa però mantenint elements obsolets. Decantar-se per la primera afavoriria una poda radical del codi (del qual podríem eliminar tot allò referent a la dualitat VARIANT ESCOLLIDA <-> V. E. NORMALITZADA, simplificant rutines que complicàvem per poder-les utilitzar en la transformació que du de l'una a l'altra i que ja només haurien d'assegurar l'aplicació reiterada de les transformacions F1, F2, F3, FS, C1, C2, C3, CS, TR i PV per poder passar de la SOLUCIÓ CREADA/COPIADA/CANÒNICA a la VARIANT ESCOLLIDA). Fer-ho per la segona és més còmode per a l'autor (que, a prop ja de les sis-centes pàgines, n'està fins al capdamunt d'emular Penèlope teixint i desteixint el programa), però una mínima objectivitat l'obligarà, malgrat tot, a atendre dues exigències de portes enfora:

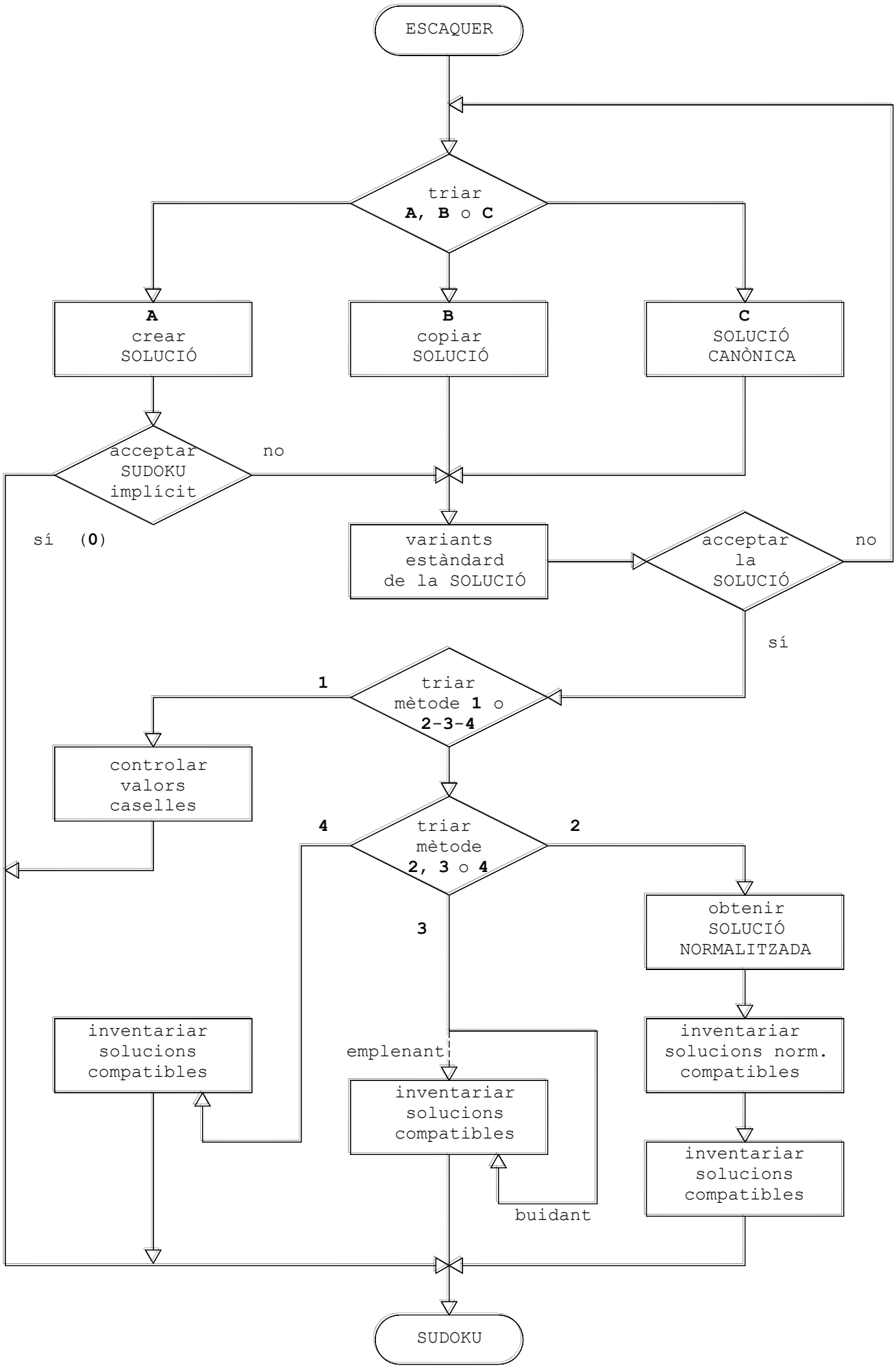
- Eliminar dels mètodes 3 i 4 qualsevol referència explícita (volem dir externa, de cara a l'usuari) a solucions normalitzades, de manera que en l'escaquer 9×9 de l'esquerra hi posi SOLUCIÓ CREADA/COPIADA/CANÒNICA (o V. E. & SUDOKU INICIAL, quan la solució provingui de l'opció A), com fins ara passava amb el mètode 1. De portes endins, si es veiés que amb els nous mètodes el codi menys enrevessat és el que cloïa el capítol precedent, atès que el muntatge preexistent responia al mètode 2 (on havíem decidit que fos ****V*V**** i no ****V**V**** la pseudomatriu 9×9 de referència), podríem limitar-nos a adaptar-lo, tot i l'absurditat conceptual de traduir la informació d'entrada a un idioma que els mètodes 3 i 4 no parlen, elaborar-la en aquesta llengua i tornar-la a traduir per donar-li sortida. O tot el contrari, almenys quan el SUDOKU-SOLUCIÓ hagués sortit de les opcions B o C: permetre que l'usuari pugués fer clic en qualsevol de les dues finestres (la de l'esquerra seria V. E. NORMALITZADA en el mètode 2 i SOLUCIÓ COPIADA/CANÒNICA en tots els demés), ampliant aquesta possibilitat al mètode 1.

- Conduir els diàlegs de tal manera que l'usuari sàpiga que hi ha dos mètodes prou eficients: l'1 (el millor, que li va donant una informació gràfica completa del progrés -emplenaments automàtics inclosos- però que, assolit el SUDOKU-PROBLEMA, ja no ofereix possibilitat de rectificació) i el 3 (que només visualitza l'eco dels propis *inputs*, però proporciona informació numèrica complementària i dóna opció a un SUDOKU-PROBLEMA alternatiu). Feta l'elecció bàsica, només si l'usuari hagués triat el mètode 3 se li oferiria la possibilitat de ratificar-la o d'anar als mètodes alternatius 2 i 4, tot advertint-lo que el 2 és el que abans brinda informació numèrica útil per moure's per l'escaquer 9×9 activant i desactivant caselles, però que també és el més afectat per llargues pauses (aquí explicaríem l'obligada subdivisió del procés en dues etapes -un mal menor necessari- i ens referiríem a les solucions normalitzades). Per tal que no se'ns veiés el llautó (convé que el consumidor cregui que el producte ofert ve a cobrir una necessitat que ignorava i respon a una lògica inapel·lable), podríem presentar els mètodes 2, 3 i 4 com una gradació creixent, definida per dues estratègies antípodes, 2 i 4, amb una virtuosa opció intermèdia 3, el punt de partida **NUM** de la qual podem aproximar al 2 (amb reserves, en prescindir de solucions normalitzades) o al 4:
  - 2: Avançar, activant caselles (amb desactivacions ocasionals), a partir del llindar d'emplenament dels SUDOKUs-PROBLEMA (**17** caselles ocupades, entre altres factors).
  - 3: Situats en un nivell d'ocupació que acceptem com a mitjana representativa dels SUDOKUs-PROBLEMA no trivials (per defecte, **NUM** = **36** caselles plenes), recular o avançar segons que hi hagi una o més solucions compatibles.
  - 4: Recular, desactivant caselles (amb activacions ocasionals), a partir d'un emplenament total o SUDOKU-SOLUCIÓ (**81** caselles ocupades).

El lector expert a llegir entre línies ja haurà clissat, darrera d'una reflexió en aparença tan ponderada, que la decisió en pro de l'opció còmoda havia estat presa: la segona, no calia dir-ho. Així queda obert el camí perquè algú, amb més aptituds de formigueta que imaginació creativa, pugui abordar la primera; i, si en aquesta empresa respecta la restricció a l'ús comercial de la llicència Creative Commons i té la noblesa de presentar-la com una derivació de la present, l'autor el beneirà. Tenint en compte que la feina ja és feta, en canvi, no fóra aconsellable dur massa lluny l'afany simplificador carregant-se, amb el mètode 2, també el 3 (el 4 ja hem dit que era una simple variant d'aquest), perquè 2, 3 i 4 tenen una particularitat de què no disposa el mètode 1: evitar l'emplenament automàtic de caselles de valor ja determinat (inhibir el dispositiu tapaforats) i fer rectificacions sense passar per la desactivació de les activacions precedents, característiques més adequades per als casos en què l'usuari busca que el SUDOKU-PROBLEMA adopti una configuració preconcebuda, pel que fa a caselles activades (en els exemples de la pàgina 423 no només perseguíem això sinó una certa ordenació numèrica, exigència que encara ens ho posava més difícil, cedint protagonisme a la fase de presolució amb l'opció A).

En descàrrec seu, però, l'autor vol adduir que la renúncia a aquesta tasca tan poc gratificant de ventafocs informàtic no només és qüestió de mandra (que també) sinó d'eficàcia, per tal de rendibilitzar el poc temps de què disposa abans no doni per acabat aquest document, si vol ser fidel a la promesa de no dedicar ni un dia més als sudokus, complerts 5 anys des de la mala hora en què va decidir de posar-s'hi. I quina feina millor que aquesta? Doncs, per exemple, revisar els fonaments de la nostra joia de la corona: un mètode 1 que funciona però que encara podem millorar.

De seguida aclarirem l'aparent estirabot, però abans d'això, a la pàgina següent, actualitzarem l'esquema de la 397, adequant-lo a la presència dels mètodes 3 i 4:


ESCAQUER
triar
A, B o C
A
crear
SOLUCIÓ
B
copiar
SOLUCIÓ
C
SOLUCIÓ
CANÒNICA
acceptar
SUDOKU
implícit
no
sí (0)
variants
estàndard
de la SOLUCIÓ
acceptar
la
SOLUCIÓ
no
sí
1
triar
mètode 1 o
2-3-4
controlar
valors
caselles
4
triar
mètode
2, 3 o 4
2
3
obtenir
SOLUCIÓ
NORMALITZADA
inventariar
solucions
compatibles
emplenant
inventariar
solucions
compatibles
buidant
inventariar
solucions norm.
compatibles
inventariar
solucions
compatibles
SUDOKU

El nou camí que havia albirat l'autor en el capítol 12 (*Un nou camí i també una drecera*) estava massa influït per la drecera (obtenció directa del SUDOKU-PROBLEMA des de la SOLUCIÓ CREADA), inspirada al seu torn en el procés de resolució espúria d'un SUDOKU-PROBLEMA utilitzant SUDÒKULUM (no sols espúria sinó absurda, com dèiem a l'inici d'aquell capítol, per bé que hagués calgut matisar que hi havia casos en que recórrer al programa podia ser legítim per estalviar-nos treball mecánic: quan arribàvem a un punt de la resolució manual, irreductible per via deductiva, en què no hi havia més sortida que verificar els dos únics valors possibles en un parell de caselles). Com que en aquell moment els arbres no el deixaven veure el bosc, va acabar donant forma a l'esguerro que coneixem com a mètode 1 (esquerro fins a cert punt, considerant la seva innegable superioritat respecte al mètode 2), fins que, enllestida pràcticament la feina i ja més distanciat de les problemàtiques locals que l'havien aclaparat fins llavors, en retornar als capítols 1 (p. 27 i 28) i 8 (p. 358 i 359) per dotar de text explicatiu uns temes només resolts pel que fa a codi de programació, va adonar-se que allò que anava escrivint no coincidia ben bé amb el que havia acabat fent: en concret ens referim a frases com "no cal estendre la comparació a totes les solucions: trobant-ne alguna altra de compatible, a més de l'original (la SUDOKU-SOLUCIÓ), ja podríem continuar", "després de l'activació, explorarem el catàleg de solucions normalitzades fins a trobar-ne alguna, diferent de l'escollida, que sigui compatibles amb la situació actual" o "escatir, després de cada activació, si només la nostra SUDOKU-SOLUCIÓ és compatible amb la part pública del sudoku (totes les caselles activades, incloent-hi l'última) o n'hi ha més (tenint en compte que, només que n'haguem trobat una altra, ja n'hi ha prou per afirmar-ho i continuar activant caselles)". Aquest és el camí que qualsevol que es posés a reflexionar sobre el tema, sense prejudicis ni apriorismes, hauria acabat fent, però no era exactament el que l'autor va emprendre en el capítol 12: no és el mateix considerar si, a més del valor que la casella que anem a activar té en la SUDOKU-SOLUCIÓ adoptada, hi ha un altre candidat vàlid (és a dir, algun altre valor que doni lloc a una solució compatible, si més no), que activar una casella i comprovar si, a més de la SUDOKU-SOLUCIÓ adoptada, hi ha alguna altra solució compatible. Com a molt podríem afirmar que tenen un cert aire de família.

Algú podria argumentar que, a la pràctica, aquest mètode, que substituirà el fins ara coneixíem com 1, prenent-li el nom, té un rendiment si fa no fa igual, perquè tant en l'antic mètode 1 com en l'actual, **RE-MEMBER** es conforma amb una solució compatible amb les caselles activades i diferent de la de referència; en el primer cas la casella que activàvem havia d'admetre un valor diferent a l'**N** de la VARIANT ESCOLLIDA; en el segon, el mateix candidat **N** havia d'admetre una solució diferent a aquesta VARIANT ESCOLLIDA. Però qui digués això estaria passant per alt una cosa fonamental: tot i tenir activades les mateixes **I** caselles i estar avaluant l'abast del nou clic (també a la mateixa casella), amb l'antic mètode exploràvem solucions compatibles amb els **I** emplenaments precedents, mentre que amb el nou n'explorem de compatibles amb **(I + 1)** emplenaments que inclouen la casella actual. Si ho voleu ras i curt, amb el nou mètode 1 anem avançats una posició. Per no fer-ho tan llarg com en els apèndixs en negreta al capítol 12, ens limitarem a considerar l'impacte de la substitució en la versió que oferíem íntegra en aquell capítol i en l'última proposada, base de la versió íntegra final que veureu ben aviat i que recull part dels propòsits enunciats dues pàgines enrera. Començar adaptant aquella primera versió, última en què **SUD-MIN** encara era accedida des de les funcions **FES-SOLUCIO**, **FES-SUDOKU-1** i **FES-SUDOKU-2** (sabem que la tercera, prou evolucionada i amb el nom **FES-SUDOKU-234**, és l'única des d'on se seguirà accedint a una **SUD-MIN** desprovista d'arguments), ens anirà bé per identificar els canvis. Fixeu-vos que en les dues primeres funcions s'accedia a **MASCARA** abans que a **ACTUALITZA** (en **FES-SUDOKU-1** no era **COMMUTA** < **CLIC** qui actualitzava ****V**V****) i en l'accés a **SUD-MIN** l'argument **PREVI** era **T**, mentre que en **FES-SUDOKU-2** s'accedia a **MASCARA** després d'actualitzar ****V*V**** i ****V**V**** (amb **COMMUTA** < **CLIC**), i en l'accés a **SUD-MIN** l'argument **PREVI** era **nil**. Així que, bàsicament, la millora del mètode 1 consistirà a introduir a **FES-SUDOKU-1** aquests canvis: passar a abans de **MASCARA** (i de **SUD-MIN**, accés en què serà **PREVI** = **nil**) l'actualització de ****V**V****, **LLL** i **LLL/LLL**. A més d'això, haurem de condicionar el final del procés a **(= M 1)**, en comptes de fer-ho a **(COMPLET-9*9 **V**V**)**, modificar el final de **FES-SUDOKU**, simplificant dràsticament **RASTRE**, i adaptar mínimament **MASCARA** i **RE-MEMBER** per tal de limitar l'exploració al valor **N** i forçar-ne la continuació quan la solució localitzada és la VARIANT ESCOLLIDA. Per facilitar la comparació amb el codi de les mateixes funcions en el capítol 12, les mostrarem senceres i subratllant, com de costum, les expressions modificades:

```
(defun RE-MEMBER (R-9*9 R-LLL / 2R R-N R-P)
  (if INI (setq 2R (ACTUALITZA R-9*9 N P R-LLL ())
                R-9*9 (car 2R) R-LLL (cadr 2R) INI ()))
```

```
  (if (not OK)
      (if (and (COMPLET-9*9 R-9*9)
               (not (and 1-2 (equal R-9*9 0-**9**9**))))
          (setq OK T)
          (if R-P
              (progn
                (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
                (foreach E L1A9
                  (if (and (not OK) (member E R-N))
                      (progn
                        (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                        (RE-MEMBER (car 2R) (cadr 2R))))))))
  OK)

(defun MASCARA (L / INI OK _ JJ 9**9 0-**9*9** 0-LLL 0-L 0-**9**9**)
  (if (not 1-2) (setq LL () M 0 SS (ssadd)))
  (if POST
      (progn
        (if 1-2 (setq 0-**9**9** (if C->F (TRANSPOSAR **9**9**) **9**9**)))
        (if C->F (TRANSPOSA-HO))
        (if (< Y 3)
            (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL)
            (progn
              (setq JJ (if (< Y 6) '(1 0 2) '(2 1 0))
                    0-LLL (F=3 (if 1-2 **V**V** **9*9**)))
              (if 1-2 (setq 0-**9**9** (F=3 0-**9**9**)))
              (setq K -1)
              (foreach EE 0-LLL
                (setq 0-L () K (1+ K))
                (foreach E EE
                  (setq 0-L (cons (if (atom E) E (list (car E) K)) 0-L)))
                (setq 0-**9*9** (cons (reverse 0-L) 0-**9*9**)))
              (setq 0-**9*9** (reverse 0-**9*9**)
                    0-LLL (F=3 LLL) P (list X (rem Y 3)))))))
  (if (and POST 1-2)
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")))
  (foreach N L1A9
    (if (and POST (not 1-2))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "CLIC sobre"
                 "el nombre..."))
    (if (and (member N L)
             (or (not POST) (setq INI (not 1-2) OK ())
                 (RE-MEMBER 0-**9*9** 0-LLL)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-2)
              (progn
                (if POST (TREU-RETOL))
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
                (command "DESIGNA" (ssadd (entlast) SS) ""))))
          (if (and POST (not 1-2)) (TREU-RETOL))))
  (if POST
      (progn (if 1-2 (command "DESHACER" "R" "UY" "ESPACIOM"))
             (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
             (setq P (list X Y)))))
```

```
(defun RASTRE ()
  (if 1-2
      (command "CVPORT" 2 "CAMBIA" "C" "0,0" "8,8" "" "P" "C" "NORM-1" ""
               "ESPACIOP" "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" "")
      (progn
        (setq ABC "D")
        (foreach E PPM (ssadd (ssname (ssget " C" (car E) (cadr E)) 0) SS))
        (command "CAMBPROP" "C" "0,0" "8,8" "E" SS "" "C" "NORM-0" ""
                 "ESPACIOP"))
  (RETOLS))

(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM) PY (cadar PPM)
            X (fix (/ (+ (car PX) (car PY)) 2))
            Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
            P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (and (not POST) (> (last (car PPM)) 0))
            (setq NOPLUS (COMMUTA NOPLUS)))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (if (not LLL) (INI-LLL))
        (setq LLL (ACTUALITZA **V**V** N P LLL T)
              **V**V** (car LLL) LLL (cadr LLL)
              LLL/LLL (cons LLL LLL/LLL) M 1 GR N)
        (if (not POST) (SUD-MIN NOPLUS ()))
        (MASCARA (list N))
        (if (> M 1) (setq N GR PPM (cons (list PX PY M) PPM))))))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     **V**V**)
  (if (= 1-2 "1") (INI-LLL) (progn (NORMTEXT-9*9) (EXPLICACIO)))
  (setq PPM () **V**V** (INI-COORDS)
        NOPLUS (if (= 1-2 "1") **V**V**)
        **V*V** (if (= 1-2 "2") **V**V**)
        GR (if (= 1-2 "1") (strcat "Sobre la VARIANT ESCOLLIDA (dreta), feu clic "
                                   "en les caselles que hagi de veure\nel jugador"
                                   " (per anul•lar els últims clics, polseu la "
                                   "tecla \"<---Backspace\"):") "."))
  (prompt (strcat "\n.\n.\n" GR))
  (if (not (and (= ABC "A") (= 1-2 "1")))
      (command "ESPACIOM" "CVPORT" 2
               "CAMBPROP" "C" "0,0" "8,8" "" "C" "NORM-0" ""))
  (command "CVPORT" 3 "CAMBPROP" "C" "0,0" "8,8" "" "C" "VAR-0" "")
  (graphscr)
  (YIN-YANG)
  (while (not (or (and (= 1-2 "1") (= M 1))
                  (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if (or (= 1-2 "2") (= (getvar "CVPORT") 3))
             (if (= (car GR) 5)
                 (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
                 (if (or (and (= 1-2 "1") PPM (equal GR '(2 8)))
                         (and (= (car GR) 3)
                              (> (getvar "CVPORT") 1)
                              (setq P (cadr GR) PX (car P) PY (cadr P))
                              (equal (list PX PY) '(4 4) 4.48)
                              (or (< (- PX (setq X (fix PX))) 0.48)
                                  (< (- (setq X (fix (1+ PX))) PX) 0.48))
                              (or (< (- PY (setq Y (fix PY))) 0.48)
                                  (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                              (or (= 1-2 "2") (listp (ELEMENT **V**V**)))
                              (PUNT)))
                     (if (= 1-2 "1") (FES-SUDOKU-1) (FES-SUDOKU-2))))))
```

```
  (if (= 1-2 "1")
      (progn
        (command "CAPA" "DES" "VAR-0" "")
        (foreach E PPM
          (if (= (last E) 0) (command "BORRA" "C" (car E) (cadr E) "")))
        (command "CAPA" "ACT" "VAR-0" "")
        (RASTRE))
      (command "ESPACIOP"))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" ""))
```

Si ens haguéssim plantat en aquesta versió, jutjant l'eficàcia de l'antic i el nou mètode 1 exclusivament pels resultats experimentals, hauríem pogut mantenir dubtes sobre la superioritat del segon. Amb proves realitzades sobre uns quants exemples, entre els quals no podia faltar la SOLUCIÓ CANÒNICA (determinada amb 24 caselles si, en lloc d'activar les 27, de baix a dalt, com a la pàgina 273, ens saltem les caselles 6,4 (**7**), 1,6 (**8**) i 3,6 (**1**), per exemple) ni la de LA VANGUARDIA del 25-02-08, en general hauríem copsat que amb el primer mètode 1 calia seguir activant caselles més enllà del SUDOKU-PROBLEMA, mentre que el segon el detectava així que apareixia en pantalla. Però aplicant el nou mètode 1, al costat d'aquesta millora, en alguns casos augmentava el compàs d'espera després de determinades activacions: en el sudoku de LA VANGUARDIA, per exemple, la 23a i última activació era seguida per una pausa d'un minut, abans de confirmar-se'ns l'obtenció del SUDOKU-PROBLEMA.

De tota manera, ja podíem suposar que aquestes inconveniències eren imputables més a la versió de SUDÒKULUM, encara molt rudimentària pel que fa a **RE-MEMBER**, que a pretesos punts febles de l'estratègia en què es fonamenta el nou mètode 1. Perquè, si comparem ambdues estratègies conceptualment, queda prou clar quina és superior:

Primer mètode 1:
- Suposant que a més del **N** predeterminat hi hagi altres valors 1... 9 compatibles en primera instància (presents a **(nth X (nth Y 0-LLL))**), allò més freqüent serà que **RE-MEMBER** n'hagi de processar infructuosament uns quants fins a trobar-ne un que també ho sigui en última instància. Però ens quedarem sense saber si només hi havia una solució compatible amb cada valor (la trobada per **RE-MEMBER**) o més.
- En particular, quan finalment resulti que només quedava **N** com a candidat viable, **RE-MEMBER** haurà hagut de processar inútilment tots els altres valors compatibles en primera instància.
- En no saber si darrera de cada candidat viable hi ha una solució o més, després d'haver ensopegat amb un candidat viable únic (que serà **N**) encara caldrà seguir activant caselles, fins que (manualment o automàtica) estiguin totes activades.

Segon mètode 1:
- No cal que **RE-MEMBER** perdi el temps amb valors compatibles en primera instància (presents a **(nth X (nth Y 0-LLL))**) però inviables, perquè per definició **N** ho és, de compatible en última instància. Però, a canvi d'assegurar-se la jugada, haurà de buscar si n'hi ha alguna altra, de solució diferent a la VARIANT ESCOLLIDA.
- Quan no en trobi cap més, no només sabrem que **N** era l'únic candidat compatible en última instància sinó que la VARIANT ESCOLLIDA és l'única solució compatible. Llevat d'aquesta darrera activació, en què **RE-MEMBER** haurà hagut d'explorar de manera exhaustiva (com ho faria **RE+MEMBER**), en les precedents l'exploració sols haurà hagut de localitzar una primera solució diferent.

I si us sembla que no queda prou clar, tornarem a la via experimental, però amb l'adaptació al nou mètode 1 de l'última de les versions de SUDÒKULUM, que ara es limita a les funcions que teniu tot seguit. Com de costum, subratllarem el codi modificat però, a diferència de l'últim bloc de codi, les mostrem fragmentàriament (amb l'extensió indispensable per contextualitzar el subratllat), perquè ja us les presentarem senceres en la versió íntegra del programa, que no trigarem a oferir:

```
(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P FIL COL)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if (or TT DEP)
```

```
                    (progn
                      (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)))
                      (setq 2R (list N (REORD R-LLL) (REORD REPLUS))
                            TN/R-L (if TT
                                       (subst 2R (cons N TT) TN/R-L)
                                       (if TN/R-L (cons 2R TN/R-L) (list 2R))))))
                (if (> COL FIL) ...)
                (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P))))))
  (if (and 1-2 (not CAMI)) (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P)))
  (if (not (or INI OK (setq OK (and (COMPLET-9*9 R-9*9)
                                    (not (and 1-2 (equal R-9*9 0-**9**9**)))))))
      (if R-P
        (progn
          (if PRIMER (setq CAMI (cons R-P CAMI)))
          (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
          (foreach E R-N
            (if (and E (not OK) (not CANVI))
                (progn
                  (if PRIMER ...)
                  (if (not CANVI)
                      (progn
                        (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                        (if (not FORA)
                          (if (and (equal (cadr 2R) NIL*NIL)
                                   (not (and 1-2 (equal (car 2R) 0-**9**9**))))
                              (setq OK T)
                              (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)

(defun MASCA (L POST / RR-9*9 RR-LLL PRIMER CAUSANT CAMI CANVI NOLLL RE-MEM)
  (setq LL () M 0 SS (ssadd))
  (foreach N L
    (if (and POST (not 1-2)) ...)
    (if (and N _ (or (not POST)
                     (setq INI (not 1-2) PRIMER T NOLLL () CANVI () OK ())
                     (progn
                       (setq RE-MEM (RE-MEMBER 0-**9*9** 0-LLL))
                       (if CANVI
                           (progn
                             (setq JJ ...)
                             ..............
                             (if 1-2 (setq 0-**9**9** (F=3 0-**9**9** T)))
                             (setq RR-9*9 (reverse 9**9))))
                       RE-MEM)))
        (progn ...)
        (if (and POST (not 1-2)) (TREU-RETOL)))))

(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP* 0-**9**9** 0-99 *P9)
  (setq N (N-3*3 (if 1-2 **V**V** **9*9**) T) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ))
  (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
      (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
  (if 1-2 (setq 0-**9**9** (if C->F (TRANSPOSAR **9**9**) **9**9**)))
  (PERMUT-JJ N)
  (setq JJ2 JJ)
  (if JJ
      (progn ...)
      (setq 0-**9*9** (if 1-2 **V**V** **9*9**) 0-LLL LLL))
  (if (and 1-2 JJ) (setq 0-**9**9** (F=3 0-**9**9** T)))
  (setq *P () P* () J -1)
  (if 1-2 (setq *P9 ()))
  (repeat 3
    (setq 0-9 () 0-L () N ())
    (if 1-2 (setq 0-99 ()))
    (repeat 3
      (setq J (1+ J) K 0)
      (foreach E (nth J 0-**9*9**) (if (or (atom E) (equal E P)) (setq K (1+ K))))
```

```
      (setq N (cons K N)
            0-9 (cons (nth J 0-**9*9**) 0-9) 0-L (cons (nth J 0-LLL) 0-L))
      (if 1-2 (setq 0-99 (cons (nth J 0-**9**9**) 0-99))))
    (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
    (if 1-2 (setq 0-99 (reverse 0-99)))
    (PERMUT-JJ N)
    (setq JJ1 (cons JJ JJ1))
    (if JJ
        (progn ...)
        (setq *P (append *P 0-9) P* (append P* 0-L)))
    (if 1-2 (setq *P9 (append *P9 (if JJ (F=3 0-99 ()) 0-99))))
  (setq JJ1 (if (equal JJ1 '(() () ())) () (reverse JJ1)) 0-**9*9** *P 0-LLL P*)
  (if 1-2 (setq 0-**9**9** *P9))
  (if (< (car Ñ) (caddr Ñ))
      (progn
        (setq INV-F T K ())
        ..................
        (setq 0-LLL K P (list (- 8 X) (cadr P)))
        (if 1-2 (setq K () 0-**9**9** (foreach E (reverse 0-**9**9**)
                                        (setq K (cons (reverse E) K))))))
  (if 1-2 ...)
  ............
  (setq P (list X Y)))

(defun RASTRE ()
  (if 1-2
      (command "CVPORT" 2 "CAMBIA" "C" "0,0" "8,8" "" "P" "C" "NORM-1" ""
               "ESPACIOP" "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" "")
      (progn
        (setq ABC "D")
        (foreach E PPM (ssadd (ssname (ssget " C" (car E) (cadr E)) 0) SS))
        (command "CAMBIA" "C" "0,0" "8,8" "E" SS "" "P" "C" "NORM-0" "L" 0 ""
                 "ESPACIOP"))
  (RETOLS))

(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM)
            ............
            P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (= (last (car PPM)) 0)
            (while (< (last (car PPM)) 1)
                   (setq PPM (cdr PPM)
                         ............
                         SS (ssget "_C" PX PY))
                   (V+- "VAR-1")))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (if (not LLL) (INI-LLL L1A9))
        (setq LLL (ACTUALITZA **V**V** N P LLL T) **V**V** (car LLL)
              LLL (cadr LLL) LLL/LLL (cons LLL LLL/LLL) GR N)
        (MASCARA (list N))
        (if (> M 0) (setq N GR PPM (cons (list PX PY M) PPM))))))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     T**V*V** **V**V** VTV TRANSP)
  (if (= 1-2 "1") ...)
  ..................
  (YIN-YANG)
  (while (not (or (and (= 1-2 "1") (= M 0))
                  (equal (setq GR (grread T)) '(2 13))
                  (equal GR '(2 32)) (= (car GR) 25)))
         (if ...))
  (if (= 1-2 "1")
      (progn
```

```
      (command "CAPA" "DES" "VAR-0" "")
      (foreach E PPM
        (if (= (last E) 0) (command "BORRA" "C" (car E) (cadr E) "")))
      (command "CAPA" "ACT" "VAR-0" "")
      (RASTRE))
    (command "ESPACIOP"))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" ""))
```

Utilitzant els mateixos exemples d'abans, constatem que el compàs d'espera entre activacions (per a una mateixa proporció de caselles activades) s'ha reduït. Ara bé, a banda d'això, els resultats no són espectaculars, però no pas perquè el nou mètode 1 no doni la talla sinó perquè, en un percentatge elevat de casos, l'antic (gràcies a unes versions molt més afinades de **RE-MEMBER** i **ACTUALITZA**) ja era capaç d'assolir els SUDOKUs-PROBLEMA sense necessitat d'activar manualment més caselles. Entre els cinc exemples d'abans, haurem de recórrer a un altre sudoku "difícil" de LA VANGUARDIA (en aquest cas, del dia 27-12-12), de 26 caselles. Emplenant-les de dalt a baix amb el nou mètode 1 s'obté directament el SUDOKU-PROBLEMA, però fent-ho amb l'antic caldrà una activació addicional (per exemple, la casella 5,8 (7)) o dues (per exemple, la casella 6,3 (7) i una més, encara) perquè es produeixi el rebobinat final que ens mostrarà el SUDOKU-PROBLEMA despullat. Curiosament, emplenant-les de baix a dalt anirem a un sudoku de 25 caselles en què la 4,9 (**9**) quedarà buida, i sobre el qual l'antic mètode 1 obligarà a una (de nou 5,8 (7), per exemple) o a dues (de nou 6,3 (7) per exemple, amb una altra) activacions addicionals.

3 6 2	**9 4 8**	**5** 7 1
1 5 8	3 7 2	**9** 4 6
4 **7** 9	5 6 1	3 **2 8**
6 1 **5**	**2** 9 **3**	7 8 4
4 7 9	7 1 4	6 5 **2**
7 2 **4**	**6** 8 **5**	**1** 3 **9**
2 9 6	8 3 7	4 **1** 5
5 4 **7**	1 2 6	8 9 3
8 3 **1**	**4 5 9**	2 6 7

Extemporàniament (a l'*Epíleg*) i amb una intervenció d'última hora (com qui diu amb l'irrupció *in extremis* d'aquell Setè de Cavalleria, com en els *westerns* clàssics), presentem l'última versió de SUDÒKULUM (l'última d'aquesta publicació), que seria adequat qualificar de "programa mínim" en la mesura que es limita a remodelar allò indispensable per mantenir el compromís enunciat per l'autor a la pàgina 578, en què, després de reconèixer que ja no li quedava esma per refer el programa de dalt a baix, proposava de mantenir els mètodes 2 i 4, i mirava de daurar la píndola amb aquesta sortida diplomàtica: "... però una mínima objectivitat l'obligarà, malgrat tot, a atendre dues exigències de portes enfora:

- Eliminar dels mètodes 3 i 4 qualsevol referència explícita (volem dir externa, de cara a l'usuari) a solucions normalitzades...
- Conduir els diàlegs de tal manera que l'usuari sàpiga que hi ha dos mètodes prou eficients: l'1 i el 3. Feta l'elecció bàsica, només si hagués triat el mètode 3 se li oferiria la possibilitat de ratificar-la o d'anar als mètodes alternatius 2 i 4, tot advertint-lo que el 2 és el que abans brinda informació numèrica útil per moure's per l'escaquer 9×9 activant i desactivant caselles, però que també és el més afectat per llargues pauses...".

Aprofitarem la intervenció cosmètica per abordar dos punts que, tot i ser foteses, convenia tractar: evitar que la desfilada de candidats en el caseller central 3×3 sigui tan parsimoniosa ja des del primer emplenament, en l'opció A d'obtenció de la SUDOKU-SOLUCIÓ (SOLUCIÓ CREADA), i permetre *a posteriori* l'esmena d'uns errors ben fàcils de cometre mentre transcrivim 81 valors en l'opció B (SOLUCIÓ COPIADA).

Sobre el segon retoc no caldrà afegir cap comentari a un codi prou explícit, però respecte al primer farem una breu digressió, i no pas perquè els canvis no siguin d'interpretació immediata, sinó perquè s'entengui el fet que no ens haguem quedat amb la solució més senzilla i que tampoc no haguem volgut anar més enllà. Haurem de començar dient que la decisió presa en en l'annex al capítol 12 *Actualització a l'última versió de FES-SOLUCIO del capítol 5- Etapa prèvia: crear una solució, III (roda el món i torna al Born...)*, en el sentit que **SUD-MIN** deixés d'intervenir en **FES-SOLUCIO** i **POST** s'activés d'entrada (perquè el perfeccionament de **RE-MEMBER** ja havia reduït a una durada admissible l'intèrval d'aparició dels candidats), es va revelar poc encertada tan bon punt la vam posar en pràctica i vam copsar l'efecte que produïa: fins i tot a l'usuari menys reflexiu l'estranyaria que els nou valors 1...9 apareguessin d'un a un (i no de cop) en activar la primera casella, en total absència de condicionaments, al marge que la cadència fos o no raonablement lenta. El primer intent va ser tan elemental com substituir, a **FES-SOLUCIO**, l'expressió

```
(MASCARA (ELEMENT LLL))
```

per

```
  (if (equal **9*9** 0*0)
      (progn
        (MASCA L1A9 ())
        (setq LL L1A9))
      (MASCARA (ELEMENT LLL)))
```

però de seguida vam veure que l'objectiu d'evitar temps morts no imprescindibles s'havia acomplert, però potser massa radicalment i tot, perquè, havent-se trobat l'usuari amb un caseller central 3×3 ple (per obra de **(MASCA L1A9 ())** < **TECLAT** < **FES-SOLUCIO**), gairebé no percebria que el caseller s'hagués buidat (en fer clic sobre la finestra de l'esquerra) i que immediatament s'hagués tornat a omplir, i es quedaria amb la sensació que el clic havia estat ignorat, tot i el desplaçament de la mira quadrada **CURSOR**. A més, no hauríem fet més que traslladar el problema a la segona activació, agreujat perquè, d'arrencar amb un ritme massa pausat, havíem anat al contrast encara més inexplicable entre la rapidesa del primer emplenament i la lentitud del segon i successius. En vista d'això vam decidir tornar a **MASCARA** perquè la determinació de **0-**9*9**** i **0-LLL**, i la recuperació del rètol *Espereu el missatge "CLIC sobre el nombre..."*, malgrat ser innecessaris, donessin al primer emplenament unes característiques singulars, de transició a les posteriors (ja amb el rètol, però amb un retard menor que aconseguiríem ampliant la condició d'accés a **RE-MEMBER** a **(if (and N (or (not POST) (equal **9*9** 0*0) ...) ...)**, en **MASCA**), però abans de posar-nos-hi vam pensar que seria millor estendre la singularitat al segon i al tercer emplenament, ja que fins al quart no calia que **ACTUALITZA-PLUS** < < **ACTUALITZA** entrés en acció a la cerca de forats per tapar. Només caldria afegir a **FES-SOLUCIO** el contador de caselles activades **CA**, inicialitzat a **1** i que aniríem incrementant d'unitat en unitat cap al final de cada iteració de **while**. Pel que fa a **MASCA**, incorporaríem l'expressió subratllada a la condició d'accés a **RE-MEMBER** **(if (and N (or (not POST) (and CA (< CA 4)) ...) ...)** per tal que els tres primers emplenaments eludissin aquest accés. Estèticament (perquè ara parlem d'estètica) havíem avançat, però encara podíem suavitzar la transició entre la immediatesa del primer emplenament i la lentitud del quart i successius. I, com que inserir pauses entre candidats fent **(repeat 499999 (* 9 9))** i **(repeat 999999 (* 9 9))** en el segon i tercer emplenament, per exemple, encara seria més artificiós, hem decidit que en el primer emplenament cap candidat vagi a **RE-MEMBER**, que en el segon sols hi vagin els valors **7**, **8** i **9** (si un d'aquests ja no figurés a **(ELEMENT LLL)**, no passa res) i que en el tercer sols hi vagin els valors **4**, **5**, **6**, **7**, **8** i **9** (si un o dos d'ells ja no figuressin a **(ELEMENT LLL)**, no passa res): únicament cal que en **FES-SOLUCIO** inicialitzem **CA** a **10** i que en l'assignació al final de cada iteració de **while** fem **(setq CA (if (> CA 4) (- CA 3)) ...)**, i que en la funció **MASCA** ampliem la condició d'accés a **RE-MEMBER**, fent **(if (and N (or (not POST) (and CA (< N CA)) ...) ...)**. I, ja sense més incisos i digressions, la segona i última versió íntegra del codi:

```
(defun *** () (terpri) (repeat 78 (prompt "*")))

(defun DIB-3*3 (Z)
  (command "SCP" "DE" "1,1"
           "RECTANG" "-1.485,-1.485" "1.485,1.485")
  (if Z (command "ZOOM" "E" "ZOOM" "0.9X"))
  (command "DESCOMP" "LT"
           "MATRIZ" "0,-1.485" "" "R" 3 1 0.99
           "MATRIZ" "-1.485,0" "" "" 1 3 0.99))

(defun TRIA-G ()
  (command "DESHACER" "M"
           "VENTANAS" "" "ESPACIOM"
;          "NAVVCUBEDISPLAY" 0                          ; Només si és ACAD 2011.
           "ZOOM" "C" "7,4.45" 15.8)
  (repeat 3
    (setq G (1+ G))
    (DIB-3*3 ())
    (command "TEXTO" "MC" "-1,-1" "0.5" "0" "1"
             "TEXTO" "MC" "0,-1" "" "" "2"
             "TEXTO" "MC" "1,-1" "" "" "3"
             "TEXTO" "MC" "-1,0" "" "" "4"
             "TEXTO" "MC" "0,0" "" "" "5"
             "TEXTO" "MC" "1,0" "" "" "6"
             "TEXTO" "MC" "-1,1" "" "" "7"
             "TEXTO" "MC" "0,1" "" "" "8"
```

```
            "TEXTO" "MC" "1,1" "" "" "9"
            "CAMBPROP" "OC" "-1,0" "0,1" "1,0" "0,-1" ""
                       "E" "C" "-0.2,-0.2" "0.2,0.2" "" "O" G ""
            "RECTANG" "1.515,-1.485" "13.485,1.485"
            "COLOR" G "SOMBREA" "S" "LT" "" "COLOR" "PORCAPA"
            "TEXTO" "MC" "7.5,1" "" "" "Cal diferenciar prou"
            "TEXTO" "MC" "7.5,0" "" "" "1-3-5-7-9 de 2-4-6-8"
            "TEXTO" "MC" "7.5,-1" "" "" "i llegir bé aquest text."
            "SCP" "DE" "-1,2.45"))
  (while (not (and (= (car (setq GR (grread))) 3)
                   (setq P (cadr GR) PX (car P) PY (cadr P))
                   (> PX -0.485) (< PX 14.485)
                   (setq G (cond ((and (> PY -10.835) (< PY -7.865)) 252)
                                 ((and (> PY  -7.385) (< PY -4.415)) 253)
                                 ((and (> PY  -3.935) (< PY -0.965)) 254))))))
  (command "DESHACER" "R" "TILEMODE" 1))

(defun PREPGRAF-9*9 ()
  (setq G 251               ; Es pot escollir entre els grisos ACI 252, 253 i 254.
        OSN (getvar "OSMODE")
        FIL (getvar "FILLMODE")
        SRT (getvar "SORTENTS")
        SVT (getvar "SAVETIME")
;       HPD (getvar "HPDRAWORDER")                          ; Només si és ACAD 2011.
;       DRA (getvar "DRAWORDERCTL")                         ; Només si és ACAD 2011.
;       LAY (getvar "LAYLOCKFADECTL")                       ; Només si és ACAD 2011.
        ECO (getvar "CMDECHO"))
  (setvar "CMDECHO" 0)
  (command "DESHACER" "I"
;          "LINEACOM"                                       ; Només si és ACAD 2011.
;          "LAYLOCKFADECTL" 0                               ; Només si és ACAD 2011.
;          "DRAWORDERCTL" 0                                 ; Només si és ACAD 2011.
;          "HPDRAWORDER" 1                                  ; Només si és ACAD 2011.
           "SAVETIME" 0
           "SORTENTS" 127
           "FILLMODE" 1
           "OSMODE" 0)
  (repeat 30 (terpri))
  (textscr)
  (***) (***)
  (prompt "\n***************  Benvingut/Benvinguda al programa SUDÒKULUM!")
  (prompt "  ****************")
  (***) (***)
  (prompt "\n\n\nSUDÒKULUM no ha estat concebut per resoldre sudokus (tot i que ")
  (prompt "pot usar-se amb\naquest propòsit, renunciant al goig de jugar-hi com ")
  (prompt "Déu mana) sinó per fer-ne.\n\nFunciona sobre AutoCAD 2004, i supera ")
  (prompt "altres productes similars en que la seva\nlentitud exasperant obliga")
  (prompt "rà l'usuari a exercitar la virtut de la paciència...\n\nConvé aclarir")
  (prompt " que SUDÒKULUM no té res a veure amb CURRÍCULUM ni amb ESPÈCULUM.\n\n")
  (repeat 68 (prompt " "))
  (prompt "Joan Colom\n\n\nAbans que res cal fer quatre ajustos (si en algun ")
  (prompt "moment surt un *Cancelar* i,\ndues línies més avall, un Comando: , ")
  (prompt "no en feu cas i seguiu les instruccions):\n\n1) ")
  (prompt "Deixeu maximitzada")
; (prompt "Amplieu manualment") ; Substitueix la precedent, només si és ACAD 2011.
  (prompt " la finestra de text [Per seguir, polseu qualsevol tecla] ")
  (grread)
  (graphscr)
  (prompt "\n.\n2) Deixeu a baix de tot 3 línies de text  ")
  (prompt "[Per seguir, polseu qualsevol tecla] ")
  (grread)
  (prompt "\n.\n3) Obriu la pàgina \"Visual.\" i en el requadre \"Elementos de ")
  (prompt "presentación\"\n   deixeu activada únicament la 1ª casella ")
  (prompt "[Per seguir, sortiu amb \"Aceptar\"]\n")
  (command "OPCIONES"
           "CAPA" "E" "PAPER" "I" "~PAPER"
;                 "C" "NOR-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "O" 7       ; ACAD 2000
                  "CR" "NORM-VAR,NORM-0,NORM-1,VAR-0,VAR-1" "CO" 7    ; ACAD 2004
```

```
                "PAPER,NORM-VAR,NORM-1,VAR-1" "D" "NORM-VAR" ""
          "COLOR" "PORCAPA" "TIPOLIN" "D" "CONTINUOUS" ""
          "LWDISPLAY" 0 "ESTILO" "STANDARD" "TXT" 0 1 0 "N" "N" "N"
          "TILEMODE" 1 "VENTANAS" "N"
          "SCP" "" "PLANTA" "" "SIMBSCP" "DES" "MODOSOMBRA" "2D"
          "PRESENTACION" "N" "SUDOKU"
          "PRESENTACION" "D" "SUDOKU" "SIMBSCP" "DES")
  (prompt "\n.\n4) Feu clic sobre una de les tres finestres: on el color GRIS ")
  (prompt "es diferencii bé\n   del BLANC i el NEGRE, i doni bon contrast entre ")
  (prompt "text i fons.     ")
  (TRIA-G)
  (DIB-3*3 T)
  (command
;         "MODOSOMBRA" "O"   ; Només si és ACAD 2000 o 2004, i és imprescindible.
          "CAMBPROP" "C" "-1,-1" "1,1" "" "O" G ""
          "MATRIZ" "V" "-1.5,-1.5" "1.5,1.5" "" "" 3 3 "0,0" "3,3"
          "CAPA" "B" "NORM-VAR"
;               "C" G "NORM-0,VAR-0" "D" "PAPER" ""                ; ACAD 2000
                "CO" G "NORM-0,VAR-0" "D" "PAPER" ""               ; ACAD 2004
          "PRESENTACION" "D" "SUDOKU"
          "VENTANAS" "-1.25,-0.5" "-0.25,0.5"
          "VENTANAS" "0.25,-0.5" "1.25,0.5"
          "ZOOM" "E" "ZOOM" "0.9X"  ; Ajusteu de manera que els retols de peu de
;         "ZOOM" "E" "ZOOM" "0.85X" ; finestra quedin folgats, en l'àrea grafica.
          "ESPACIOM"
;         "CVPORT" 2                                               ; ACAD 2000
          "CVPORT" 2 "UCSVP" 0                                     ; ACAD 2004
          "VGCAPA" "I" "VAR-1,VAR-0" "" ""
          "ZOOM" "-1.515,-1.515" "7.515,7.515"
;         "NAVVCUBEDISPLAY" 0                          ; Només si és ACAD 2011.
;         "CVPORT" 3                                               ; ACAD 2000
          "CVPORT" 3 "UCSVP" 0                                     ; ACAD 2004
          "VGCAPA" "I" "NORM-1,NORM-0" "" ""
          "ZOOM" "-1.515,-1.515" "7.515,7.515"
;         "NAVVCUBEDISPLAY" 0                          ; Només si és ACAD 2011.
          "SCP" "" "ESPACIOP"
          "VENTANAS" "B" "ACT" "-0.25,0.5" "0.25,-0.5" ""))

(defun PREPTEXT-9*9 ()
  (textscr)
  (prompt "\n\nAquest programa permet crear un SUDOKU-PROBLEMA a partir d'una ")
  (prompt "SUDOKU-SOLUCIÓ,\na base de decidir quines caselles es faran públiques")
  (prompt " (visibles per al jugador).\nPel que fa a la SUDOKU-SOLUCIÓ:")
  (prompt "\n\nA) Podeu improvisar-la, emplenant casella a casella sota control ")
  (prompt "del programa.")
  (prompt "\n\nB) Podeu adoptar-ne una de coneguda, que caldrà transcriure ")
  (prompt "casella a casella.")
  (prompt "\n\nC) Si la solució us és indiferent podeu usar aquesta, que ")
  (prompt "anomenarem CANÒNICA:")
  (prompt "\n\n                              9 1 2  3 4 5  6 7 8")
  (prompt   "\n                              6 7 8  9 1 2  3 4 5")
  (prompt   "\n                              3 4 5  6 7 8  9 1 2")
  (prompt "\n\n                              8 9 1  2 3 4  5 6 7")
  (prompt   "\n                              5 6 7  8 9 1  2 3 4")
  (prompt   "\n                              2 3 4  5 6 7  8 9 1")
  (prompt "\n\n                              7 8 9  1 2 3  4 5 6")
  (prompt   "\n                              4 5 6  7 8 9  1 2 3")
  (prompt   "\n                              1 2 3  4 5 6  7 8 9")
  (initget "A B C")
  (setq ABC (getkword "\n\nTrieu l'opció A, B o C <C>: ") ABC (if ABC ABC "C"))
  (graphscr))

(defun RETOL-1 (P R / C)
  (setq C (getvar "CLAYER"))
  (if (wcmatch R "V`.*") (command "BORRA" "C" "-0.85,-0.58" "-0.65,-0.52" ""))
  (command "CAPA" "D" "PAPER" ""
           "TEXTO" "MC" P 0.05 0 R
           "CAPA" "D" C ""))
```

```
(defun RETOL-2 (T1 T2 T3 T4 T5 T6 T7 T8 T9)
  (setq K 0.5)
  (foreach E (list T1 T2 T3 T4 T5 T6 T7 T8 T9)
    (RETOL-1 (list 0 (setq K (- K 0.1))) E))
; (command "ESPACIOM" "ESPACIOP")                          ; Només si és ACAD 2011.
)

(defun ACT-9*9 (N P L1 / L2)
  (reverse (foreach F L1 (setq L2 (cons (if (member P F) (subst N P F) F) L2)))))

(defun ACT-LLL (X Y A-LLL / X/3 Y/3 LL L J I)
  (setq X/3 (/ X 3) Y/3 (/ Y 3) J -1)
  (foreach F A-LLL
    (setq L () I -1 J (1+ J))
    (foreach E F
      (setq I (1+ I)
            L (cons (if (= J Y)
                        (if (= I X) () (subst () A-N E))
                        (if (= I X)
                            (subst () A-N E)
                            (if (and (= (/ I 3) X/3) (= (/ J 3) Y/3))
                                (subst () A-N E) E)))
                    L)))
    (setq LL (cons (reverse L) LL)))
  (reverse LL))

(defun P1P2 (/ P1 P2)
  (if 1-3
      (progn
        (setq P1 (mapcar '- A-P TG) P2 (mapcar '+ A-P TG)
              PPM (cons (list P1 P2 0) PPM)
              LLL/LLL (cons (ACT-LLL (car A-P) (cadr A-P) (car LLL/LLL))
                            LLL/LLL))
        (command "COPIA" "C" P1 P2 "" "0,0" ""
                 "CAMBIA" "LT" "" "P" "C" "VAR-1" "L" 1 ""))

      (command "ESPACIOM" "TEXTO" "MC" A-P 0.5 0 (itoa A-N))))

(defun DEPURA (/ E)
  (foreach N (reverse E-L) (setq E (cons (if (member N L-DEP) () N) E))))

(defun ACT-DEPURA (/ XX YY)
  (if (not XY)
      (if (not (equal (setq XX (caar P-DEP)) (caadr P-DEP)))
          (setq XX () YY (cadar P-DEP))))
  (mapcar '(lambda (F-9 F-L)
             (mapcar '(lambda (E-9 E-L)
                        (if E-L
                            (if (member E-9 P-DEP)
                                E-L
                                (if (and XY (equal E-9 XY 1.1))
                                    (DEPURA)
                                    (if XX
                                        (if (= (car E-9) XX)
                                            (DEPURA)
                                            E-L)
                                        (if (and YY (= (cadr E-9) YY))
                                            (DEPURA)
                                            E-L))))))
                     F-9 F-L))
          A-9*9 A-LLL))

(defun PLUS-N->P (M)
  (setq A-N Ñ A-P M PLUS T)
  (if REAL
      (P1P2)
      (if INI (setq REPLUS (ACT-9*9 Ñ M REPLUS)))))
```

```
(defun ATZUCAC (E)
  (if (and PRIMER (not INI) E (> (/ (cadr E) 3) 0)
           (setq I (car CAMI))
           (not (or (= (cadr E) (cadr I))
                    (and (= (/ (car E) 3) (/ (car I) 3))
                         (= (/ (cadr E) 3) (/ (cadr I) 3))))))
      (setq CAUSANT I NOLLL E))
  (setq FORA T))

(defun PRESENTS ()
  (foreach E F
    (foreach M (nth E F-L) (if (not (member M L)) (setq L (cons M L))))))

(defun EXPL-SUBCONJ (/ L OK)
  (setq K 1)
  (foreach Ñ (nth (- (length F-L) 4) LLISTES)
    (if (not (or PLUS FORA))
        (progn
          (setq K (1+ K))
          (foreach F Ñ
            (if (not (or PLUS FORA))
                (progn
                  (setq L ())
                  (if (< (1+ K) (length F-L))
                      (progn
                        (PRESENTS)
                        (if (= (length L) K)
                            (progn
                              (setq P-DEP () OK ())
                              (foreach E F (setq P-DEP (cons (nth E F-9) P-DEP)))
                              (foreach M L
                                (if (not OK)
                                    (mapcar '(lambda (E-9 E-L)
                                               (if (and (not OK)
                                                        (not (member E-9 P-DEP))
                                                        (member M E-L))
                                                   (setq OK T L-DEP L PLUS T)))
                                            F-9 F-L)))
                              (if (and FIL/COL (not OK))
                                  (if (= FIL/COL 1)
                                      (setq FIL (+ FIL K))
                                      (setq COL (+ COL K))))))))
                  (if (and (not FORA) (> K 2))
                      (progn
                        (if (not L) (PRESENTS))
                        (if (< (length L) K) (setq FORA T)))))))))))

(defun EXPLORA-9+9+5 (FIL/COL / NN MN F-9 F-L)
  (foreach Ñ L1A9 (if (not (member Ñ F-9*9)) (setq NN (cons Ñ NN))))
  (if (> (length NN) 0)
      (progn
        (setq NN (reverse NN) MN (last NN))
        (if (/= FIL/COL 2)
            (foreach E F-9*9 (if (and (not NM) (listp E)) (setq NM E))))
        (foreach Ñ NN
          (if (not PLUS)
              (progn
                (setq K 0)
                (if (and INI (= Ñ MN)) (setq F-9 () F-L ()))
                (mapcar '(lambda (E-9*9 E-LL3)
                           (if (member Ñ E-LL3) (setq K (1+ K) J E-9*9))
                           (if (and INI E-LL3 (= Ñ MN))
                               (progn
                                 (setq I ())
                                 (foreach M E-LL3 (if M (setq I (cons M I))))
                                 (setq F-9 (cons E-9*9 F-9) F-L (cons I F-L)))))
                        F-9*9 F-LL3)
```

```
                (if (= K 1)
                    (PLUS-N->P J)
                    (if (= K 0)
                        (ATZUCAC NM)
                        (if (and INI (= Ñ MN) (> (length F-L) 3))
                            (EXPL-SUBCONJ)))))))))

(defun TRANSPOSAR (A / LLE LL LE E LS LLS)
  (setq LLE A)
  (while LLE
         (setq LS () LL LLE)
         (foreach L LL (setq E (car L) LE (cdr L)
                             LS (append LS (list E))
                             LLE (if (not (equal LLE LL)) LLE)
                             LLE (if LE (append LLE (list LE)))))
         (setq LLS (append LLS (list LS)))))

(defun SUB-X*Y (X*Y P1 P2 / J K L LL)
  (setq K -1)
  (foreach Y X*Y
    (setq K (1+ K) J -1)
    (if (not (or (< K (cadr P1)) (> K (cadr P2))))
        (progn
          (foreach X Y
            (setq J (1+ J))
            (if (not (or (< J (car P1)) (> J (car P2))))
                (setq L (cons X L))))
          (setq LL (cons (reverse L) LL) L ()))))
  (reverse LL))

(defun SUPR (E L)
  (append (reverse (cdr (member E (reverse L)))) (cdr (member E L))))

(defun TRANSP-3*3 (L / X Y E EE EEE)
  (setq X 3)
  (repeat 3
    (setq X (1- X) Y 3 EE ())
    (repeat 3
      (setq E (nth X (nth (setq Y (1- Y)) L)) EE (cons E EE)))
    (setq EEE (cons EE EEE))))

(defun 3*3/R-L (/R-L)
  (setq I (list (list X Y) R-L)
        /R-L (if /R-L
                 (if (setq J (assoc N /R-L))
                     (subst (cons N (cons I (cdr J))) J /R-L)
                     (cons (list N I) /R-L))
                 (list (list N I)))))

(defun TERCETS (<R-9 <R-L / R-9 R-L TRANSP TERC SEGUIR I J K K1 K2)
  (foreach 3N 3*N
    (if (and (not TERC) (member N 3N))
        (progn
          (setq R-9 <R-9 R-L <R-L
                N1 (car 3N) N2 (cadr 3N) N3 (last 3N)
                TRANSP ())
          (repeat 2
            (if (not TERC)
                (progn
                  (setq I () J -1 K 0)
                  (mapcar '(lambda (F/R-9 F/R-L)
                             (setq J (1+ J) K1 0 K2 0)
                             (mapcar '(lambda (E/R-9 E/R-L)
                                        (if (and (or E/R-L (= N1 E/R-9)
                                                     (= N2 E/R-9) (= N3 E/R-9))
                                                 (or (not E/R-L) (member N1 E/R-L)
                                                     (member N2 E/R-L)
                                                     (member N3 E/R-L)))
```

```
                                        (setq K1 (1+ K1))
                                        (setq K2 (1+ K2))))
                                  F/R-9 F/R-L)
                             (if (= K1 3) (setq I J))
                             (if (= K2 3) (setq K (1+ K))))
                        R-9 R-L)
                  (if (and I (= K 2))
                      (progn
                        (foreach L (setq J (nth I R-L))
                          (setq K1 ())
                          (foreach E (setq K L)
                            (if (and E (not (member E 3N)))
                                (setq TERC T K1 T K (subst () E K))))
                          (if K1 (setq J (subst K L J))))
                        (if TERC (setq R-L (subst J (nth I R-L) R-L)))
                        (setq J -1 K1 () SEGUIR T)
                        (foreach N (reverse L1A9)
                          (if (not (member N 3N)) (setq K1 (cons N K1))))
                        (while (and SEGUIR (< J 3))
                          (setq K2 K1 J (1+ J))
                          (if (/= J I)
                              (progn
                                (mapcar '(lambda (E/R-9 E/R-L)
                                           (if E/R-L
                                               (foreach N E/R-L
                                                 (if (member N K2)
                                                     (setq K2 (SUPR N K2))))
                                               (setq K2 (SUPR E/R-9 K2))))
                                        (nth J R-9) (nth J R-L))
                                (if (= (length K2) 3)
                                    (progn
                                      (setq SEGUIR ())
                                      (foreach L (setq K (nth (- 3 (+ I J)) R-L)
                                                       K1 K)
                                        (if L
                                            (progn
                                              (setq I () J I)
                                              (foreach E L
                                                (setq J
                                                      (cons
                                                        (if E
                                                            (if (member E K2) E
                                                                (not (setq I T))))
                                                        J)))
                                              (if I (setq K1 (subst (reverse J)
                                                                    L K1))))))
                                      (if (not (equal K1 K))
                                          (setq R-L (subst K1 K R-L) TERC T)))))))
                        (if TERC (setq R-L (if TRANSP (TRANSP-3*3 R-L) R-L)
                                       TN/R-L (3*3/R-L TN/R-L)))))
                  (if (not (or TRANSP TERC))
                      (setq R-9 (TRANSP-3*3 R-9)
                            R-L (TRANSP-3*3 R-L) TRANSP T)))))))))

(defun ACTUALITZA-PLUS (/ I L M Ñ X1 Y1 5P R-9 R-L ORIG)
  (setq FORA ())
  (mapcar '(lambda (9*9 LL3)
             (setq ORIG (not ORIG))
             (if (not (or PLUS FORA))
                 (mapcar '(lambda (F-9*9 F-LL3)
                            (if (and (not (or PLUS FORA)) ORIG)
                                (mapcar '(lambda (E-9*9 E-LLL)
                                           (if (and (not (or PLUS FORA)) E-LLL)
                                               (progn
                                                 (setq K 0)
                                                 (foreach E E-LLL
                                                   (if (and (< K 2) E)
                                                       (setq Ñ E K (1+ K))))
```

```
                                                    (if (= K 1)
                                                        (PLUS-N->P E-9*9)
                                                        (if (= K 0)
                                                            (ATZUCAC E-9*9))))))
                                            F-9*9 F-LL3))
                                    (if (not (or PLUS FORA))
                                        (EXPLORA-9+9+5 (if ORIG 1 2))))
                             9*9 LL3)))
          (list A-9*9 (TRANSPOSAR A-9*9)) (list A-LLL (TRANSPOSAR A-LLL)))
  (if (not (or PLUS FORA))
      (progn
        (setq X1 (* (/ X 3) 3) Y1 (* (/ Y 3) 3) ORIG ())
        (foreach K '(6 3 0) (if (/= K Y1) (setq 5P (cons (list X1 K) 5P))))
        (foreach K '(6 3 0) (setq 5P (cons (list K Y1) 5P)))
        (repeat (if INI 2 1)
          (setq ORIG (not ORIG))
          (if (not (or PLUS FORA))
              (foreach Y '(0 3 6)
                (foreach X '(0 3 6)
                  (if (and ORIG (not (or PLUS FORA)) (member (list X Y) 5P))
                      (progn
                        (setq XY (list (1+ X) (1+ Y))
                              R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                  (+ Y 2)))
                              5P (subst (cons (list X Y) (list R-9 R-L))
                                                         (list X Y) 5P))
                        (mapcar '(lambda (F-9*9 F-LL3)
                                   (if (not (or PLUS FORA)) (EXPLORA-9+9+5 ())))
                                (list (append (car R-9) (cadr R-9) (last R-9)))
                                (list (append (car R-L) (cadr R-L) (last R-L))))))
                  (if (not ORIG)
                      (progn
                        (if (setq R-L (assoc (list X Y) 5P))
                            (setq R-9 (cadr R-L) R-L (caddr R-L))
                            (setq R-9 (SUB-X*Y A-9*9 (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))
                                  R-L (SUB-X*Y A-LLL (list X Y) (list (+ X 2)
                                                                      (+ Y 2)))))
                        (setq K 0)
                        (foreach F R-9
                          (foreach E F (if (atom E) (setq K (1+ K)))))
                        (if (< K 9) (TERCETS R-9 R-L))
                        (setq N/R-L (3*3/R-L N/R-L)))))))))))

(defun ACTUALITZA (A-9*9 A-N A-P A-LLL REAL / X Y XY L PLUS P-DEP L-DEP)
  (if (and REAL (setq TT (cdr (assoc A-N TN/R-L))))
      (progn
        (setq A-9*9 (ACT-9*9 A-N A-P A-9*9) A-LLL (car TT) L (cadr TT) K -1)
        (foreach F L
          (setq J -1 K (1+ K))
          (foreach E F
            (setq J (1+ J))
            (if (atom E)
                (progn
                  (setq A-P (list J K) A-N E A-9*9 (ACT-9*9 A-N A-P A-9*9))
                  (P1P2))))))
      (while (not PLUS)
             (setq X (car A-P) Y (cadr A-P) FIL 0 COL 0)
             (if L-DEP
                 (setq A-LLL (ACT-DEPURA) DEP T L-DEP () XY ())
                 (setq A-9*9 (ACT-9*9 A-N A-P A-9*9)
                       A-LLL (ACT-LLL X Y A-LLL) XY ()))
             (ACTUALITZA-PLUS)
             (setq PLUS (not PLUS))))
  (if REAL (setq N/R-L () TN/R-L ()))
  (list A-9*9 A-LLL))
```

```
(defun APLEGA (P1 P2 P3 / L1 L2 L3)
  (setq L1 (assoc P1 TT) L1 (cadr (if L1 L1 (assoc P1 K)))
        L2 (assoc P2 TT) L2 (cadr (if L2 L2 (assoc P2 K)))
        L3 (assoc P3 TT) L3 (cadr (if L3 L3 (assoc P3 K))))
  (list (append (car  L1) (car  L2) (car  L3))
        (append (cadr L1) (cadr L2) (cadr L3))
        (append (last L1) (last L2) (last L3))))

(defun RE-ACT-LLL (/ K)
  (setq K (cdr (assoc N N/R-L))
        K (append (APLEGA '(0 0) '(3 0) '(6 0))
                  (APLEGA '(0 3) '(3 3) '(6 3))
                  (APLEGA '(0 6) '(3 6) '(6 6)))))

(defun FILS/COLS (/ TRANSP SEGUIR I J K L LL M M1 3/R-9 3/R-L)
  (repeat 2
    (if TRANSP (setq R-9*9 (TRANSPOSAR R-9*9) R-LLL (TRANSPOSAR R-LLL)))
    (setq M1 () M ())
    (mapcar '(lambda (F/R-9 F/R-L)
               (setq K1 () K2 L1A9 K -2)
               (repeat 3
                 (setq K (+ K 3) SEGUIR T LL ()
                       3/R-9 (list (nth (1- K) F/R-9) (nth K F/R-9)
                                   (nth (1+ K) F/R-9))
                       3/R-L (list (nth (1- K) F/R-L) (nth K F/R-L)
                                   (nth (1+ K) F/R-L)))
                 (mapcar '(lambda (E/R-9 E/R-L)
                            (if SEGUIR
                                (progn
                                  (if E/R-L
                                      (progn
                                        (setq L ())
                                        (foreach E E/R-L
                                          (if (and E (not (member E LL)))
                                              (setq L (cons E L))))
                                        (if (> (length L) 3)
                                            (setq SEGUIR ())
                                            (setq LL (append L LL))))
                                      (setq LL (cons E/R-9 LL)))
                                  (if (and SEGUIR (> (length LL) 3))
                                      (setq SEGUIR ())))))
                          3/R-9 3/R-L)
                 (if SEGUIR
                     (progn
                       (foreach N K2 (if (member N LL) (setq K2 (SUPR N K2))))
                       (setq K1 (append K1 (list 3/R-L))))
                     (setq I 3/R-L)))
               (if (and (= (length K1) 2) (= (length K2) 3))
                   (progn
                     (setq J ())
                     (foreach L I
                       (setq J (cons (if L (progn
                                             (setq K ())
                                             (foreach E (reverse L)
                                               (setq K (cons (if (member E K2) E)
                                                             K)))))
                                     J)))
                     (setq J (reverse J)
                           F/R-L (cond ((equal F/R-L
                                               (append I (car K1) (cadr K1)))
                                        (append J (car K1) (cadr K1)))
                                       ((equal F/R-L
                                               (append (car K1) I (cadr K1)))
                                        (append (car K1) J (cadr K1)))
                                       (T (append (car K1) (cadr K1) J))))))
               (setq M1 (if M (append M1 (list M))) M F/R-L))
             R-9*9 R-LLL)
```

```
    (setq R-LLL (append M1 (list M)) TRANSP (not TRANSP)))
  (setq R-9*9 (TRANSPOSAR R-9*9) R-LLL (TRANSPOSAR R-LLL)))

(defun INVERT-JJ (KK / II)
  (foreach E '(2 1 0) (setq II (cons (- 3 (length (member E KK))) II))))

(defun F=3 (9*9 9*3 / 9**9)
  (foreach J JJ
    (if 9*3
        (progn
          (setq K (1- (* 3 J)))
          (repeat 3 (setq K (1+ K) 9**9 (cons (nth K 9*9) 9**9))))
        (setq 9**9 (cons (nth J 9*9) 9**9))))
  (reverse 9**9))

(defun REORD (K / I J 3J JJ)
  (if INV-F (foreach F (reverse K) (setq J (cons (reverse F) J))) (setq J K))
  (if JJ1
      (progn
        (setq K () I -1)
        (repeat 3
          (setq 3J ())
          (repeat 3
            (setq I (1+ I))
            (setq 3J (cons (nth I J) 3J)))
          (setq 3J (reverse 3J)
                JJ (nth (/ I 3) JJ1)
                JJ (if JJ (INVERT-JJ JJ))
                K (append K (if JJ (F=3 3J ()) 3J)))))
      (setq K J))
  (if JJ2
      (setq JJ (INVERT-JJ JJ2) J (F=3 K T))
      (setq J K))
  (if C->F (TRANSPOSAR J) J))

(defun COMPLET-9*9 (9*9 / COMP)
  (setq COMP T)
  (foreach L 9*9
    (if COMP
        (foreach E L
          (if (and COMP (listp E)) (setq COMP () R-P (if POST E))))))
  COMP)

(defun RE-MEMBER (R-9*9 R-LLL / REPLUS FORA DEP 2R R-N R-P FIL COL)
  (if INI
      (progn
        (setq REPLUS 0*0 2R (ACTUALITZA R-9*9 N P R-LLL ()) INI FORA)
        (if (not INI)
            (progn
              (setq R-9*9 (car 2R) R-LLL (cadr 2R)
                    TT (cdr (assoc N TN/R-L)))
              (if (or TT DEP)
                  (progn
                    (if TT (setq R-LLL (RE-ACT-LLL) R-LLL (FILS/COLS)))
                    (setq 2R (list N (REORD R-LLL) (REORD REPLUS))
                          TN/R-L (if TT
                                     (subst 2R (cons N TT) TN/R-L)
                                     (if TN/R-L (cons 2R TN/R-L) (list 2R)))))
              (if (> COL FIL)
                  (setq R-9*9 (TRANSPOSAR R-9*9)
                        R-LLL (TRANSPOSAR R-LLL) K ()
                        R-9*9 (foreach F (reverse R-9*9)
                                (setq J ())
                                (foreach E (reverse F)
                                  (setq J (cons (if (atom E) E (reverse E)) J)))
                                (setq K (cons J K)))))
              (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P))))))
  (if (and 1-3 (not CAMI)) (setq RR-9*9 R-9*9 RR-LLL R-LLL CAMI (list P)))
```

```
  (if (not (or INI OK (setq OK (and (COMPLET-9*9 R-9*9)
                                    (not (and 1-3 (equal R-9*9 0-**9**9**)))))))
      (if R-P
        (progn
          (if PRIMER (setq CAMI (cons R-P CAMI)))
          (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
          (foreach E R-N
            (if (and E (not OK) (not CANVI))
                (progn
                  (if PRIMER
                      (progn
                        (setq CAMI (member R-P CAMI))
                        (if (and NOLLL (not (member CAUSANT CAMI)))
                            (setq CANVI T))))
                  (if (not CANVI)
                      (progn
                        (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                        (if (not FORA)
                          (if (and (equal (cadr 2R) NIL*NIL)
                                   (not (and 1-3 (equal (car 2R) 0-**9**9**))))
                              (setq OK T)
                              (RE-MEMBER (car 2R) (cadr 2R)))))))))))
  OK)

(defun TREU-RETOL () (repeat 4 (command "BORRA" "LT" "")))

(defun MASCA (L POST / RR-9*9 RR-LLL PRIMER CAUSANT CAMI CANVI NOLLL RE-MEM)
  (setq LL () M 0 SS (ssadd))
  (foreach N L
    (if (and POST (not 1-3))
        (RETOL-2 "Espereu" "el missatge" "" "" "" "" "" "\"CLIC sobre"
                 "el nombre...\""))
    (if (and N (or (not POST) (and CA (< N CA))
                   (setq INI (not 1-3) PRIMER T NOLLL () CANVI () OK ())
                   (progn
                     (setq RE-MEM (RE-MEMBER 0-**9*9** 0-LLL))
                     (if CANVI
                         (progn
                           (setq JJ (if (= (/ (cadr NOLLL) 3) 1)
                                        '(1 0 2)
                                        '(2 0 1))
                                 9**9 () I -1)
                           (foreach EE (F=3 RR-9*9 T)
                             (setq 0-L () I (1+ I))
                             (foreach E EE
                               (setq 0-L (cons (if (atom E) E (list (car E) I))
                                               0-L)))
                             (setq 9**9 (cons (reverse 0-L) 9**9)))
                           (if 1-3 (setq 0-**9**9** (F=3 0-**9**9** T)))
                           (setq RR-9*9 (reverse 9**9) RR-LLL (F=3 RR-LLL T)
                                 PRIMER () CANVI ()
                                 RE-MEM (RE-MEMBER RR-9*9 RR-LLL))))
                     RE-MEM)))
        (progn
          (setq LL (cons N LL) M (1+ M))
          (if (not 1-3)
              (progn
                (if POST (TREU-RETOL))
                (RETOL-1 (cond ((= N 1) '(-0.15 0.15))
                               ((= N 2)     '(0 0.15))
                               ((= N 3)  '(0.15 0.15))
                               ((= N 4) '(-0.15 0))
                               ((= N 5)     '(0 0))
                               ((= N 6)  '(0.15 0))
                               ((= N 7) '(-0.15 -0.15))
                               ((= N 8)     '(0 -0.15))
                               (   T     '(0.15 -0.15)))
                         (itoa (if MASC (nth (1- N) MASC) N)))
```

```
                (command "DESIGNA" (ssadd (entlast) SS) ""))))
        (if (and POST (not 1-3)) (TREU-RETOL)))))

(defun TECLAT ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.2,-0.2" "-0.1,-0.1"
           "COPIA" "LT" "" "-1.0419,0.5919" ""
           "EDITPOL" (setq CURSOR (ssget "_L")) "G" 0.006 ""
           "MATRIZ" "-0.1,-0.1" "" "R" 3 3 0.15 0.15
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "COLOR" G
           "SOMBREA" "S" "C" "-0.22,-0.53" "0.22,0.53" ""
           "COLOR" "PORCAPA")
  (MASCA L1A9 ())
  (if (and (not RESP) (not 1-3))
      (command "ESPACIOM" "CVPORT" 2 "CAPA" "D" "NORM-1" "")))

(defun INI-LLISTES (/ LA LL)
  (foreach N '(3 4 5 6 7 8 9)
    (setq LA ())
    (repeat N (setq N (1- N) LA (cons (list N) LA)))
    (setq LL LA)
    (repeat (1- (length LL))
      (setq LA LL LL ())
      (repeat (1- (length LA))
        (setq I (car LA) LA (cdr LA) J (reverse (cdr (reverse I))))
        (foreach E LA
          (if (equal (reverse (cdr (reverse E))) J)
              (setq LL (cons (append I (list (last E))) LL)))))
      (setq LL (reverse LL) LLL (cons LL LLL)))
    (setq LLL (reverse (cdr LLL)))
    (repeat (- (length (last LLL)) 3) (setq LLL (cdr LLL)))
    (setq LLISTES (cons LLL LLISTES)))
  (setq LLISTES (cdr (reverse LLISTES))))

(defun INI-LLL (L)
  (setq LL () LLL ())
  (repeat 9 (setq LL (cons L LL)))
  (repeat 9 (setq LLL (cons LL LLL))))

(defun INI-COORDS ()
  (setq Y -1 PY ())
  (reverse (repeat 9 (setq Y (1+ Y) X -1 PX ()
                           PY (cons (reverse (repeat 9 (setq X (1+ X)
                                                             PX (cons (list X Y)
                                                                      PX))))
                                    PY)))))

(defun ELEMENT (LL) (nth X (nth Y LL)))

(defun TECLA-M (/ FI-B)
  (while (not (or (setq GR (grread)
                        FI-B (and (= ABC "B") (or (equal GR '(2 13))
                                                  (equal GR '(2 32))
                                                  (= (car GR) 25))))
                  (and (= (car GR) 3)
                       (= (getvar "CVPORT") 2)
                       (setq P (cadr GR) PX (car P) PY (cadr P))
                       (equal (list PX PY) '(4 4) 4.48)
                       (or (< (- PX (setq X (fix PX))) 0.48)
                           (< (- (setq X (fix (1+ PX))) PX) 0.48))
                       (or (< (- PY (setq Y (fix PY))) 0.48)
                           (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                       (or (= ABC "B") (ELEMENT LLL))))))
  (if (not FI-B) (setq P (list X Y))))

(defun M->P () (list (+ (* 0.1105 X) -1.1919) (+ (* 0.1105 Y) -0.4419)))
```

```
(defun TRANSPOSA-HO ()
  (setq K (if 1-3 **V**V** **9*9**)
        K (TRANSPOSAR K) LLL (TRANSPOSAR LLL) P (reverse P) 9**9 ())
  (foreach EE K
    (setq 0-L ())
    (foreach E EE (setq 0-L (cons (if (atom E) E (reverse E)) 0-L)))
    (setq 9**9 (cons (reverse 0-L) 9**9)))
  (setq 9**9 (reverse 9**9) K X X Y Y K)
  (if 1-3 (setq **V**V** 9**9) (setq **9*9** 9**9)))

(defun PERMUT-JJ (NUM)
  (setq JJ (if (and (>= (car NUM) (cadr NUM))
                    (>= (cadr NUM) (last NUM)))
               ()
               (if (and (<= (car NUM) (cadr NUM))
                        (<= (cadr NUM) (last NUM)))
                   '(2 1 0)
                   (if (>= (car NUM) (last NUM))
                       (if (> (cadr NUM) (last NUM))
                           '(1 0 2)
                           '(0 2 1))
                       (if (> (cadr NUM) (last NUM))
                           '(1 2 0)
                           '(2 0 1)))))))

(defun N-3*3 (9*9 P-SEMPRE / 3*3 *3 I)
  (setq K -1)
  (repeat 3
    (setq *3 ())
    (repeat 3
      (setq J -1 K (1+ K))
      (repeat 3
        (setq N 0)
        (repeat 3
          (setq J (1+ J) I (nth J (nth K 9*9)))
          (if (or (atom I) (and P-SEMPRE (equal I P))) (setq N (1+ N))))
        (setq *3 (cons N *3))))
    (setq *3 (list (+ (nth 2 *3) (nth 5 *3) (nth 8 *3))
                   (+ (nth 1 *3) (nth 4 *3) (nth 7 *3))
                   (+ (nth 0 *3) (nth 3 *3) (nth 6 *3))) 3*3 (cons *3 3*3)))
  (setq 3*3 (reverse 3*3)))

(defun 1+2+3 (LL) (mapcar '(lambda (L) (eval (cons '+ L))) LL))

(defun MASCARA (L / C->F INV-F JJ JJ1 JJ2 N/R-L INI OK Ñ 9**9 0-**9*9** 0-9 0-LLL
                    0-L *P P* *PP* 0-**9**9** 0-99 *P9)
  (setq N (N-3*3 (if 1-3 **V**V** **9*9**) T) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ))
  (if (< (eval (cons 'max N)) (eval (cons 'max Ñ)))
      (progn (TRANSPOSA-HO) (setq K N N Ñ Ñ K C->F T)))
  (if 1-3 (setq 0-**9**9** (if C->F (TRANSPOSAR **9**9**) **9**9**)))
  (PERMUT-JJ N)
  (setq JJ2 JJ)
  (if JJ
      (progn
        (setq 0-LLL () I -1)
        (foreach EE (F=3 (if 1-3 **V**V** **9*9**) T)
          (setq 0-L () I (1+ I))
          (foreach E EE (setq 0-L (cons (if (atom E) E (list (car E) I)) 0-L)))
          (setq 0-LLL (cons (reverse 0-L) 0-LLL)))
        (setq 0-**9*9** (reverse 0-LLL) 0-LLL (F=3 LLL T)
              P (list X (+ (rem Y 3) (* 3 (- 3 (length (member (/ Y 3) JJ))))))))
      (setq 0-**9*9** (if 1-3 **V**V** **9*9**) 0-LLL LLL))
  (if (and 1-3 JJ) (setq 0-**9**9** (F=3 0-**9**9** T)))
  (setq *P () P* () J -1)
  (if 1-3 (setq *P9 ()))
  (repeat 3
    (setq 0-9 () 0-L () N ())
```

```
    (if 1-3 (setq 0-99 ()))
    (repeat 3
      (setq J (1+ J) K 0)
      (foreach E (nth J 0-**9*9**) (if (or (atom E) (equal E P)) (setq K (1+ K))))
      (setq N (cons K N)
            0-9 (cons (nth J 0-**9*9**) 0-9) 0-L (cons (nth J 0-LLL) 0-L))
      (if 1-3 (setq 0-99 (cons (nth J 0-**9**9**) 0-99))))
    (setq N (reverse N) 0-9 (reverse 0-9) 0-L (reverse 0-L))
    (if 1-3 (setq 0-99 (reverse 0-99)))
    (PERMUT-JJ N)
    (setq JJ1 (cons JJ JJ1))
    (if JJ
        (progn
          (setq *PP* () I (- J 3))
          (if (and (> (cadr P) I) (<= (cadr P) J))
              (setq P (list X (1+ (- J (length (member (rem (cadr P) 3) JJ)))))))
          (foreach EE (F=3 0-9 ())
            (setq 0-9 () I (1+ I))
            (foreach E EE (setq 0-9 (cons (if (atom E) E (list (car E) I)) 0-9)))
            (setq *PP* (cons (reverse 0-9) *PP*)))
          (setq *P (append *P (reverse *PP*)) P* (append P* (F=3 0-L ())))
        (setq *P (append *P 0-9) P* (append P* 0-L)))
    (if 1-3 (setq *P9 (append *P9 (if JJ (F=3 0-99 ()) 0-99)))))
  (setq JJ1 (if (equal JJ1 '(() () ())) () (reverse JJ1)) 0-**9*9** *P 0-LLL P*)
  (if 1-3 (setq 0-**9**9** *P9))
  (if (< (car Ñ) (caddr Ñ))
      (progn
        (setq INV-F T K ())
        (foreach EE (reverse 0-**9*9**)
          (setq 0-L ())
          (foreach E EE
            (setq 0-L (cons (if (atom E) E (list (- 8 (car E)) (cadr E))) 0-L)))
          (setq K (cons 0-L K)))
        (setq 0-**9*9** K K ())
        (foreach E (reverse 0-LLL) (setq K (cons (reverse E) K)))
        (setq 0-LLL K P (list (- 8 X) (cadr P)))
        (if 1-3 (setq K () 0-**9**9** (foreach E (reverse 0-**9**9**)
                                       (setq K (cons (reverse E) K)))))))
  (if 1-3
      (progn
        (command "ESPACIOP" "BORRA" "LT" ""
                 "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
        (RETOL-2 "Hi ha" "un procés" "de cerca" "en marxa." ""
                 "Per seguir," "espereu" "el símbol" "del TAO...")
;       (command "ESPACIOM" "ESPACIOP")                    ; Només si és ACAD 2011.
      ))
  (MASCA L T)
  (if 1-3 (command "DESHACER" "R" "UY" "ESPACIOM"))
  (if C->F (progn (TRANSPOSA-HO) (setq C->F ())))
  (setq P (list X Y)))

(defun TECLA-P (VARS / FI)
  (while (not (and (or (= (car (setq GR (grread))) 3)
                       (setq FI (and VARS (> (strlen MSG) 51)
                                     (or (equal GR '(2 13)) (equal GR '(2 32))))))
                   (or FI (equal (setq GR (cadr GR)) GR (list (car GR) (cadr GR)))
                                 '(0 0) 0.2))
                   (or FI (setq N (cond ((equal GR '(-0.15  0.15) 0.05) 1)
                                        ((equal GR '( 0     0.15) 0.05) 2)
                                        ((equal GR '( 0.15  0.15) 0.05) 3)
                                        ((equal GR '(-0.15  0   ) 0.05) 4)
                                        ((equal GR '( 0     0   ) 0.05) 5)
                                        ((equal GR '( 0.15  0   ) 0.05) 6)
                                        ((equal GR '(-0.15 -0.15) 0.05) 7)
                                        ((equal GR '( 0    -0.15) 0.05) 8)
                                        ((equal GR '( 0.15 -0.15) 0.05) 9))))
                   (or VARS (= ABC "B") (member N LL))))))
```

```
(defun REPESCA ()
  (command "DESPLAZA" CURSOR "" "0,0.9942" "" "ESPACIOM")
  (setq P-CURS '(-1.1919 0.4419))
  (prompt "\n.\n.\nOPCIÓ A ESMENAR ELS VALORS EQUIVOCATS.")
  (while (progn
           (prompt "\n.\nA l'esquerra, CLIC sobre la casella a esmenar ")
           (prompt "(<Intro> o < > per acabar): ")
           (TECLA-M))
         (command "ESPACIOP"
                  "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P))
                  "ESPACIOM"
                  "BORRA" "C" (mapcar '- P TG) (mapcar '+ P TG) ""
                  "ESPACIOP")
         (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
         (TECLA-P ())
         (command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
         (setq **9**9** () K -1)
         (foreach F **9*9**
           (if (= (setq K (1+ K)) Y)
               (progn
                  (setq L () J -1)
                  (foreach C F
                    (if (= (setq J (1+ J)) X) (setq C N))
                    (setq L (cons C L)))
                  (setq F (reverse L))))
           (setq **9**9** (cons F **9**9**)))
         (setq **9*9** (reverse **9**9**) **9**9** ())))

(defun RETOLS ()
  (RETOL-1 (list (if 1-3 0.75 -0.75) -0.55) "SUDOKU-PROBLEMA")
  (RETOL-1 (list (if 1-3 -0.75 0.75) -0.55) "SUDOKU-SOLUCIÓ"))

(defun RASTRE ()
  (if 1-3
      (command "CVPORT" 2 "BORRA" "C" "0,0" "8,8" ""
               "CVPORT" 3 "CAPA" "DES" "VAR-1" ""
               "COPIA" "C" "0,0" "8,8" "" "0,0" ""
               "CAMBIA" "P" "" "P" "C" "NORM-1" ""
               "DESIGNA" "C" "0,0" "8,8" ""
               "CAPA" "ACT" "VAR-1" ""
               "ORDENAOBJETOS" "P" "" "T")
      (progn
        (setq ABC "D")
        (foreach E PPM (ssadd (ssname (ssget "_C" (car E) (cadr E)) 0) SS))
        (command "CAMBIA" "C" "0,0" "8,8" "E" SS "" "P" "C" "NORM-0" "L" 0 ""
                 "ESPACIOP")
        (RETOLS))))

(defun FES-SOLUCIO (/ CURSOR P-CURS CA)
  (TECLAT)
  (if (= ABC "A")
      (progn
        (INI-LLISTES)
        (INI-LLL ())
        (setq CA 10 PPM () NIL*NIL LLL
              0*0 (INI-COORDS) **9*9** 0*0
              P-CURS '(-1.1919 0.4419) POST T)
        (INI-LLL L1A9)
        (while (not (COMPLET-9*9 **9*9**))
               (prompt "\n.\n.\nA l'esquerra, CLIC sobre la casella a omplir: ")
               (TECLA-M)
               (command "ESPACIOP" "BORRA" SS ""
                        "DESPLAZA" CURSOR "" P-CURS (setq P-CURS (M->P)))
               (MASCARA (ELEMENT LLL))
               (prompt "\nCLIC sobre el nombre que va a la casella marcada: ")
               (TECLA-P ())
               (command "ESPACIOM" "TEXTO" "MC" P 0.5 0 (itoa N))
```

```
              (setq CA (if (> CA 4) (- CA 3))
                    PPM (cons (list (mapcar '- P TG) (mapcar '+ P TG) M) PPM)
                    LLL (ACTUALITZA **9*9** N P LLL T)
                    **9*9** (car LLL) LLL (cadr LLL)))
        (while (= (last (car PPM)) 1) (setq PPM (cdr PPM)))
        (setq POST () SS (ssadd))
        (prompt "\n.\nPodeu seguir, per construir el SUDOKU-PROBLEMA (el joc) ")
        (prompt "a partir d'aquesta\nSUDOKU-SOLUCIÓ, o acabar i quedar-vos amb ")
        (prompt "el que heu definit sobre la marxa.")
        (initget "Si No")
        (setq CONF (getkword "\nVoleu seguir (S/N)? <S>: ")))
      (progn
        (command "ESPACIOP")
        (prompt "\n.\n.\nCLIC sobre el nombre que va a la casella marcada: ")
        (setq K 9)
        (repeat 9
          (setq L () J -1 K (1- K))
          (repeat 9
            (setq J (1+ J))
            (TECLA-P ())
            (command "DESPLAZA" CURSOR "" "0.110465,0" "" "ESPACIOM"
                     "TEXTO" "MC" (list J K) 0.5 0 (itoa N) "ESPACIOP")
            (setq L (append L (list N))))
          (command "DESPLAZA" CURSOR "" "-0.9942,-0.1105" "")
          (setq **9*9** (cons L **9*9**)))
        (REPESCA)))
  (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
           "CAMBPROP" "P" "" "C" "VAR-1" "")
  (if (= CONF "No") (RASTRE) (command "ESPACIOP"))
  (command "BORRA" CURSOR "C" "-0.22,-0.53" "0.22,0.53" ""))

(defun FES-+9*9+- (/ -+9*9+- E F)
  (setq -+9*9+- **9*9**)
  (foreach P PPM
    (setq X (fix (/ (+ (caar P)  (caadr P))  2))
          Y (fix (/ (+ (cadar P) (cadadr P)) 2))
          F (nth Y -+9*9+-) E (nth X F)
          -+9*9+- (subst (subst (- E) E F) F -+9*9+-))))

(defun GRAF-9*9 (VAR / I)
  (setq J -1)
  (foreach 9*9 (list **9*9** **9**9**)
    (if (< J 0)
        (command "CVPORT" 3 "BORRA" "C" "0,0" "8,8" ""
                 "CVPORT" 2 "BORRA" "C" "0,0" "8,8" ""
                 "CAPA" "D" (if VAR "NORM-0" "NORM-1") "")
        (progn
          (setq VAR () J -1)
          (command "CVPORT" 3 "CAPA" "D" "VAR-1" "")))
    (while (< (setq J (1+ J)) 9)
           (setq K -1)
           (while (< (setq K (1+ K)) 9)
                  (setq I (nth J (nth K 9*9)))
                  (command "TEXTO" "MC" (list J K) 0.5 0 (itoa (abs I)))
                  (if (and VAR (< I 0))
                      (command "CAMBPROP" "LT" "" "C" "NORM-1" ""))))))

(defun SEGUIR (GRAF)
  (if GRAF (prompt "\n.\n."))
  (prompt "\n                 [Per seguir, polseu qualsevol tecla]")
  (while (/= (car (grread)) 2)))

(defun VARIACIONS ()
  (textscr)
  (repeat 16 (terpri))
  (prompt "D'aquesta SUDOKU-SOLUCIÓ se'n poden treure d'altres, per aplicació ")
  (prompt "reiterada\nde les transformacions següents:")
  (prompt "\n\n\nF1: Permutació de Files d'una mateixa tríada\n    (6 ")
```

```
  (prompt "permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).")
  (prompt   "\n\nF2: Permutació de Files de tríades diferents")
  (prompt "\n    (fins a 216 permutacions, però no sempre és possible).")
  (prompt   "\n\nF3: Permutació de tríades de Files (6 permutacions).")
  (prompt "\n\n\nFS: Simetria respecte a la Fila central (es pot fer amb tres ")
  (prompt "F1 i una F3).")
  (prompt "\n\n\nC1: Permutació de Columnes d'una mateixa tríada\n    (6 ")
  (prompt "permutacions per tríada -> 216 possibilitats, fent-ho en totes tres).")
  (prompt "  \n\nC2: Permutació de Columnes de tríades diferents")
  (prompt "\n    (fins a 216 permutacions, però no sempre és possible).")
  (prompt   "\n\nC3: Permutació de tríades de Columnes (6 permutacions).")
  (prompt "\n\n\nCS: Simetria respecte a la Columna central (es pot fer amb tres")
  (prompt " C1 i una C3).")
  (prompt "\n\n\nTR: Transposició de files i columnes.")
  (prompt "\n\n\nPV: Permutació entre Valors 1 a 9 estesa a tot el SUDOKU (tot i")
  (prompt " que muntat com\n    una seqüència de permutacions binàries, hi ha 9!")
  (prompt " = 362.880 possibilitats).\n\n")
  (SEGUIR ()))

(defun AREES (/ G)
  (setq K -1)
  (if (wcmatch E "*column*")
      (if (< F 3)
          (repeat 9 (setq K (1+ K)
                          G (list (mapcar '- (list K 0) TG)
                                  (mapcar '+ (list K 8) TG))
                          GG (cons G GG)))
          (repeat 3 (setq K (1+ K)
                          G (list (mapcar '- (list (* 3 K) 0) TG)
                                  (mapcar '+ (list (+ 2 (* 3 K)) 8) TG))
                          GG (cons G GG))))
      (if (< F 3)
          (repeat 9 (setq K (1+ K)
                          G (list (mapcar '- (list 0 K) TG)
                                  (mapcar '+ (list 8 K) TG))
                          GG (cons G GG)))
          (repeat 3 (setq K (1+ K)
                          G (list (mapcar '- (list 0 (* 3 K)) TG)
                                  (mapcar '+ (list 8 (+ 2 (* 3 K))) TG))
                          GG (cons G GG)))))
  (setq GG (reverse GG)))

(defun PREVIES ()
  (setq E (if (= (substr RESP 1 1) "F") "fil" "column")
        F (atoi (substr RESP 2 1))
        E (if (= F 1) (strcat E "a") (if (= F 2) E (strcat "tríada de " E "es"))))
  (AREES))

(defun ZONA (P)
  (setq Z () K -1)
  (foreach G GG
    (if (and (not Z) (setq K (1+ K))
             (< (caar G) (car P))  (< (cadar G) (cadr P))
             (< (car P) (caadr G)) (< (cadr P) (cadadr G)))
        (setq Z K)))
  Z)

(defun CONTROL (/ OK)
  (if ZZ
      (if (= Z (car ZZ))
          (setq OK ())
          (if (= F 3)
              (setq OK T)
              (progn
                (setq OK (= F 1))
                (if (or (and (= I 0) (< 2 Z))
                        (and (= I 6) (< Z 6))
                        (and (= I 3) (or (< Z 3) (< 5 Z))))
```

```
                    (setq OK (= F 2))))))
      (setq OK T I (if (< F 3) (* 3 (/ Z 3)) 0)))
  OK)

(defun INPUNT (MSG / GR Z)
  (prompt MSG)
  (while (not (and (= (car (setq GR (grread))) 3)
                   (ZONA (cadr GR))
                   (CONTROL)))
         (if Z (prompt (strcat "\nNo s'hi val>" (substr MSG 2)))))
  (setq ZZ (cons Z ZZ)))

(defun PERMUT-N (A)
  (cond ((equal A '(0 1 2)) 0)
        ((equal A '(0 2 1)) 1)
        ((equal A '(1 0 2)) 2)
        ((equal A '(2 0 1)) 3)
        ((equal A '(1 2 0)) 4)
        (   T   5)))

(defun PERMUT-G (/ L)
  (setq L (mapcar '(lambda (Z) (- Z I)) ZZ))
  (cond ((or (equal L '(1 2 1)) (equal L '(2 1 2)))
         '(0 2 1))
        ((or (equal L '(0 1 0)) (equal L '(1 0 1)))
         '(1 0 2))
        ((or (equal L '(0 1 2)) (equal L '(1 2 0)) (equal L '(2 0 1)))
         '(1 2 0))
        ((or (equal L '(0 2 1)) (equal L '(1 0 2)) (equal L '(2 1 0)))
         '(2 0 1))
        ((or (equal L '(0 2 0)) (equal L '(2 0 2)))
         '(2 1 0))))

(defun TR-9**9 (F/C)
  (setq **9**9** (if (= (substr RESP 1 1) (if F/C "F" "C"))
                    **9**9**
                    (TRANSPOSAR **9**9**))))

(defun CAN->VAR (/ 9**9)
  (TR-9**9 T)
  (if (= F 1)
      (progn
        (setq 9**9 **9**9**)
        (foreach J '(0 1 2)
          (setq K (+ I J) **9**9** (subst (nth J JJ) (nth K 9**9) **9**9**)))
        (foreach J '(0 1 2)
          (setq K (+ I J) **9**9** (subst (nth K 9**9) J **9**9**))))
      (setq **9**9** (F=3 **9**9** T)))
  (TR-9**9 T))

(defun EXECUTA (FUN / 9*9)
  (if (= ABC "A")
      (progn
        (setq 9*9 **9**9** **9**9** **9*9**)
        (FUN)
        (setq **9*9** **9**9** **9**9** 9*9)))
  (FUN))

(defun F/C (/ E F GG JJ ZZ)
  (PREVIES)
  (while (not (setq ZZ ()
                    JJ (INPUNT (strcat T1 E T2))
                    JJ (INPUNT T3)
                    JJ (INPUNT (strcat T4 E T5))
                    JJ (PERMUT-G)))
         (prompt "\nNo has definit cap permutació. TORNA-HI!"))
  (EXECUTA CAN->VAR))
```

```
(defun EXPLORA (/ BCA CAB EFD FDE HIG IGH LLL)
  (TR-9**9 T)
  (setq LL ())
  (foreach I '(0 1 2 3 4 5)
    (setq L (nth I **9**9**)
          BCA (list (nth 1 L) (nth 2 L) (nth 0 L))
          CAB (list (nth 2 L) (nth 0 L) (nth 1 L))
          EFD (list (nth 4 L) (nth 5 L) (nth 3 L))
          FDE (list (nth 5 L) (nth 3 L) (nth 4 L))
          HIG (list (nth 7 L) (nth 8 L) (nth 6 L))
          IGH (list (nth 8 L) (nth 6 L) (nth 7 L))
          LLL (list (append BCA EFD HIG) (append BCA EFD IGH)
                    (append BCA FDE HIG) (append CAB FDE IGH)
                    (append CAB FDE HIG) (append CAB EFD IGH)))
    (foreach J '(3 4 5 6 7 8)
       (if (>= J (* (1+ (/ I 3)) 3))
           (foreach K LLL
             (if (equal K (nth J **9**9**)) (setq LL (cons (list I J) LL)))))))
  (TR-9**9 T)
  (setq LL (reverse LL)))

(defun PERM-BIN ()
  (TR-9**9 T)
  (setq J (nth (car ZZ) **9**9**) K (nth (cadr ZZ) **9**9**)
        **9**9** (subst () J **9**9**)
        **9**9** (subst J  K **9**9**)
        **9**9** (subst K () **9**9**))
  (TR-9**9 T))

(defun F//C (/ E F GG JJ ZZ)
  (PREVIES)
  (prompt "\n.\n.\nAquesta VARIANT ESCOLLIDA ")
  (if (EXPLORA)
      (progn
        (prompt (strcat "permet permutar les " E "es"))
        (foreach L LL
          (prompt (strcat " " (itoa (1+ (car L))) "ª i " (itoa (1+ (cadr L))) "ª"
                          (if (equal L (last LL))
                                       "."
                                       (if (equal L (cadr (reverse LL)))
                                           ", i" ",")))))
        (while (not (setq ZZ ()
                          ZZ (INPUNT (strcat T1 "primera " E "a: "))
                          ZZ (INPUNT (strcat T1 "segona "  E "a: "))
                          ZZ (if (< (car ZZ) (cadr ZZ))
                          ZZ (reverse ZZ))
                          JJ (member ZZ LL)))
                 (prompt (strcat "\nAquesta permutació (" (itoa (1+ (car ZZ))) "-"
                                 (itoa (1+ (cadr ZZ))) ") no és permesa. TORNA-HI!")))
        (EXECUTA PERM-BIN))
      (prompt (strcat "no permet permutar " E "es de tríades diferents.\n"))))

(defun SIM ()
  (TR-9**9 ())
  (foreach F **9**9** (setq **9**9** (subst (reverse F) F **9**9**)))
  (TR-9**9 ()))

(defun INVALOR (MSG) (prompt MSG) (setq N ()) (TECLA-P T) N)

(defun 2-9V (/ CURSOR MASC N1 N2 N-F N-9*9)
  (TECLAT)
  (command "BORRA" CURSOR "")
  (setq MASC L1A9)
  (while (INVALOR (strcat "\n.\n." T6 "a substituir" (if N1 " <sortir>" "") ": "))
         (setq N1 (nth (1- N) MASC)
               N2 (nth (1- (INVALOR (strcat T6 "que el substituirà: "))) MASC))
         (if (/= N1 N2)
             (progn
```

```
            (command "BORRA" SS "")
            (setq MASC (subst N2 0 (subst N1 N2 (subst 0 N1 MASC))))
            (MASCA MASC ())
            (foreach 9*9 (if (= ABC "A")
                             (list **9*9** **9**9**)
                             (list **9**9**))
              (setq N-9*9 ())
              (foreach F 9*9
                (setq N-F ())
                (foreach E F
                  (setq I (/ E (abs E))
                        N-F (cons (if (= (abs E) N1)
                                      (* N2 I)
                                      (if (= (abs E) N2) (* N1 I) E))
                                  N-F)))
                (setq N-9*9 (cons (reverse N-F) N-9*9)))
              (setq N-9*9 (reverse N-9*9))
              (if (= ABC "A")
                  (if (equal 9*9 **9*9**)
                      (if (equal 9*9 **9**9**)
                          (setq **9**9** N-9*9)
                          (setq **9*9** N-9*9))
                      (setq **9**9** N-9*9))
                  (setq **9**9** N-9*9)))
            (command "ESPACIOM")
            (GRAF-9*9 (= ABC "A"))
            (command "ESPACIOP"))))
  (command "BORRA" "C" "-0.22,-0.53" "0.22,0.53" "" "ESPACIOM"))

(defun CANON->VARIANT (/ T1 T2 T3 T4 T5 T6 RESP OK)
  (graphscr)
  (setq T1 "\n   Assenyaleu (dreta) la " T2 " que vulgueu moure: "
        T3 "\n   Assenyaleu (dreta) on voleu situar-la: "
        T4 "\n   Assenyaleu (dreta) on voleu situar la "
        T5 " desplaçada: "
        T6 "\n   CLIC (centre) sobre el nombre " RESP "")
  (while RESP
         (if (/= (substr (getvar "LASTPROMPT") 1 35)
                 "Aquesta VARIANT ESCOLLIDA no permet")
             (prompt "\n."))
         (initget "F1 F2 F3 FS C1 C2 C3 CS TR PV")
         (setq RESP (getkword (strcat "\n.\nTipus de permutació: F1/F2/F3/FS/"
                                      "C1/C2/C3/CS/TR/PV <sortir>: ")))
         (if (not OK)
             (progn
               (if (= ABC "A")
                   (progn
                     (RETOL-1 '(-0.75 -0.55) "V. E. & SUDOKU INICIAL")
                     (setq **9*9** (FES-+9*9+-))))
               (setq OK T)
               (command "ESPACIOM")
               (if (= ABC "A") (GRAF-9*9 T))))
         (if (or (= RESP "F1") (= RESP "F3") (= RESP "C1") (= RESP "C3"))
             (F/C)
             (if (or (= RESP "F2") (= RESP "C2"))
                 (F//C)
                 (if (or (= RESP "FS") (= RESP "CS"))
                     (EXECUTA SIM)
                     (if (= RESP "TR")
                         (setq **9*9** (if (= ABC "A")
                                           (TRANSPOSAR **9*9**) **9*9**)
                               **9**9** (TRANSPOSAR **9**9**))
                         (if (= RESP "PV") (2-9V))))))
         (GRAF-9*9 (= ABC "A")))
  (initget "Si No")
  (setq CONF
        (getkword "\n.\n.\nUs sembla bé com a solució de treball (S/N)? <S>: ")
        CONF (if CONF CONF "Si")))
```

```
(defun TRIA-METODE ()
  (textscr)
  (repeat 40 (terpri))
  (prompt "Disposeu de quatre mètodes per obtenir un SUDOKU-PROBLEMA d'una ")
  (prompt "SUDOKU-SOLUCIÓ\n(aquesta VARIANT ESCOLLIDA), activant algunes ")
  (prompt "caselles (deixant-les visibles):\n\n")
  (prompt "1 (el recomanat): A partir de les caselles que empleneu, impedirà ")
  (prompt "emplenaments\n                  redundants. Només reculant podreu ")
  (prompt "desactivar les activades.")
  (prompt "\n\n2, 3 o 4 (lents): Podreu activar qualsevol casella i desactivar-")
  (prompt "ne en un ordre\n                  qualsevol, però l'únic ajut serà ")
  (prompt "conèixer quantes solucions\n                  compatibles queden ")
  (prompt "després de cada activació o desactivació.\n                  Només ")
  (prompt "recomanables si voleu dibuixar amb les caselles formes\n         ")
  (prompt "         preconcebudes, i no us importa que n'hi hagi de redundants.")
  (terpri) (terpri)
  (initget "1 3")
  (setq 1-3
        (getkword "Preferiu l'opciò 1 o alguna de les altres tres? (<1> o 3): "))
  (if (< 1-3 "3")
      (setq 1-3 "1")
      (progn
        (prompt "\n\n\n\n\nDiferència entre els mètodes 2, 3 i 4:\n\n")
        (prompt "2) El recompte de solucions compatibles pot començar amb 17 ")
        (prompt "caselles plenes.\n   Com que el nombre inicial seria molt alt, ")
        (prompt "dividirem el procés en 2 parts:\n   - En la 1ª només es tindran")
        (prompt " en compte solucions normalitzades compatibles\n     amb la V. ")
        (prompt "E. NORMALITZADA, situada a l'esquerra de la VARIANT ESCOLLIDA\n")
        (prompt "     (solucions en què les files de cada tríada estan ordenades")
        (prompt " numèricament\n     de baix a dalt i les tríades estan ordenade")
        (prompt "s atenent la fila inferior),\n     fins que sols en quedi una.")
        (prompt "\n   - En la 2ª es tindran en compte totes les solucions compa")
        (prompt "tibles, fins que\n     sols en quedi una, moment en què tindrem")
        (prompt " el SUDOKU-PROBLEMA.\n   Si el primer recompte trigués massa, ")
        (prompt "millor que cancel•leu i proveu el 3.\n\n")
        (prompt "3) Podeu escollir amb quantes caselles activades ha de començar")
        (prompt " el recompte.\n   Si no ho encerteu i el recompte no comença ")
        (prompt "amb el SUDOKU-PROBLEMA, millor\n   errar per excés que per ")
        (prompt "defecte, perquè desactivar caselles és més ràpid.\n\n")
        (prompt "4) Comença amb la SUDOKU-SOLUCIÓ plena, i cal anar desactivant ")
        (prompt "caselles fins\n   que passeu d'una a més solucions compatibles.")
        (prompt " Només és recomanable si heu\n   fet curt, usant el mètode 3, i")
        (prompt " no us fa mandra desactivar força caselles.")
        (initget "2 3 4")
        (setq 1-3 (getkword "\n\nTrieu l'opció 2, 3 o 4 <3>: ")
              1-3 (if 1-3 1-3 "3"))
        (if (= 1-3 "3")
            (progn
              (while
                (or (< (setq K (getint (strcat "\n¿Quantes caselles (entre 18 "
                                               "i 54) voleu emplenar, pel cap "
                                               "baix? <" (itoa NUM) ">: "))
                               K (if K K NUM)) 18)
                    (> K 54))
                (prompt (strcat "\n   El nombre de caselles no pot ser " (itoa K)
                                ": repetiu!")))
              (setq NUM K))))))

(defun ORDENA-3 ()
  (setq A (caar L) B (caadr L) C (caaddr L)
        D (assoc (max A B C) L) A (assoc (min A B C) L)
        L (subst () D (subst () A L))))

(defun VAR->NORM (/ K L 9**9 A B C D)
  (setq **9**9** (TRANSPOSAR **9**9**))
  (setq K -1)
```

```
  (repeat 3
    (setq L ())
    (repeat 3 (setq K (1+ K) L (cons (nth K **9**9**) L)))
    (ORDENA-3)
    (foreach E L (if E (setq L (list A E D))))
    (setq 9**9 (append 9**9 (list L))))
  (setq K -1 L ())
  (foreach E 9**9 (setq L (cons (cons (caar E) E) L)))
  (ORDENA-3)
  (foreach E L (if E (setq **9*9** (append (cdr A) (cdr E) (cdr D)))))
  (if (not T**9*9**) (setq T**9*9** **9*9**)))

(defun PRE-NORM->VAR (/ L* L** N F F1 F2 F3 F4)
  (foreach L  (reverse **9*9**) (setq L*  (cons (car L) L*)))
  (foreach L (reverse **9**9**) (setq L** (cons (car L) L**)))
  (setq N1 (- 9 (length (member (nth 0 L*) L**))) F1 (/ N1 3)
        N2 (- 9 (length (member (nth 3 L*) L**))) F2 (/ N2 3)
        N3 (- 9 (length (member (nth 6 L*) L**))) F3 (/ N3 3)
        N4 (list F1 F2 F3) N4 (PERMUT-N N4)
        F1 (rem N1 3)
        F2 (- 9 (length (member (nth 1 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 2 L*) L**))) F3 (rem F3 3)
        N1 (list F1 F2 F3) N1 (PERMUT-N N1)
        F1 (rem N2 3)
        F2 (- 9 (length (member (nth 4 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 5 L*) L**))) F3 (rem F3 3)
        N2 (list F1 F2 F3) N2 (PERMUT-N N2)
        F1 (rem N3 3)
        F2 (- 9 (length (member (nth 7 L*) L**))) F2 (rem F2 3)
        F3 (- 9 (length (member (nth 8 L*) L**))) F3 (rem F3 3)
        N3 (list F1 F2 F3) N3 (PERMUT-N N3))
  (if (not TN1) (setq TN1 N1 TN2 N2 TN3 N3 TN4 N4)))

(defun EXPLICACIO ()
  (graphscr)
  (GRAF-9*9 ())
  (command "ESPACIOP")
  (RETOL-1 '(-0.75 -0.55) "V. E. NORMALITZADA")
  (prompt "\n\nCal tenir l'arxiu 123456789.txt a la ruta de cerca d'AutoCAD. ")
  (prompt "Si no l'hi teniu,\npodeu copiar-lo en el directori d'aquest dibuix ")
  (prompt "i després seguir com si no res.")
  (SEGUIR ())
  (textscr)
  (repeat 30 (terpri))
  (prompt "Per construir el SUDOKU-PROBLEMA a partir d'aquesta ")
  (prompt "SUDOKU-SOLUCIÓ, haureu de\nseguir els passos següents:")
  (prompt "\n\n1) Com que els valors quedaran velats (en gris), haureu d'anar ")
  (prompt "fent clic a les\n   caselles que vulgueu convertir en públiques ")
  (prompt "(visibles per al jugador). Els\n   clics poden fer-se sobre la ")
  (prompt "V. E. NORMALITZADA o sobre la VARIANT ESCOLLIDA\n   (esquerra o ")
  (prompt "dreta) però per passar de l'una a l'altra caldrà un altre clic.")
  (prompt "\n\n2) Les caselles públiques quedaran destacades (blanc/negre) ")
  (prompt "en ambdues bandes,\n   però, si hi torneu a fer un clic, ")
  (prompt "passaran de nou a ser secretes (en gris).")
  (prompt "\n\n3) Quan el conjunt de caselles públiques assoleixi les ")
  (prompt "condicions  necessàries\n   per a una solució única (haver-n'hi ")
  (prompt "almenys 17; que cap tríada de files no\n   en tingui 2 de buides; ")
  (prompt "que cap tríada de columnes no en tingui 2 de buides)\n   hi haurà ")
  (prompt "la pausa més llarga. El text inferior mostrarà que hi ha activitat")
  (prompt "\n   i, quan aquesta cessi, podreu llegir-ne el resum i saber ")
  (prompt "quantes SOLUCIONS\n   NORMALITZADES, compatibles amb la ")
  (prompt "V. E. NORMALITZADA, s'han detectat.")
  (prompt "\n\n4) Si n'hi ha més d'una, haureu de seguir fent clic ")
  (prompt "fins a tenir-ne només una.")
  (prompt "\n\n5) Una nota us advertirà que el fet que sols quedi una solució ")
  (prompt "normalitzada no\n   exclou que hi hagi més solucions, ")
  (prompt "no normalitzades però també compatibles,\n   i anunciarà una segona ")
  (prompt "pausa per calcular i fer-vos saber quantes n'hi ha.")
```

```
  (prompt "\n\n6) Si hi ha més d'una solució, caldrà seguir fent clic fins que ")
  (prompt "només en quedi\n   una, que serà la normalitzada.\n")
  (SEGUIR ()))

(defun FOTO-V*V (/ FOTO)
  (setq J -1)
  (foreach L VTV
    (setq I -1 J (1+ J))
    (foreach E L
      (setq I (1+ I))
      (if (atom E) (setq FOTO (cons (list I J) FOTO)))))
  FOTO)

(defun YIN-YANG ()
  (command "ESPACIOP" "CAPA" "D" "PAPER" ""
           "RECTANG" "-0.25,-0.5" "0.25,0.5"
           "ARCO" "0,-0.15" "0.15,0" "0,0.15" "ARCO" "" "0,-0.15"
           "ARCO" "0,0" "0.075,0.075" "0,0.15"
           "ARCO" "0,0" "-0.075,-0.075" "0,-0.15"
           "CIRCULO" "0,0.075" 0.015 "CIRCULO" "0,-0.075" 0.02
           "COLOR" G  "SOMBREA" "S" "-0.15,0" "0.15,0" ""
           "BLOQUE" "C" "0,0" "LT" ""
           "SOMBREA" "S" "0,0.5" "-0.15,0" "0.15,0" "" "BORRA" "LT" ""
           "COLOR" 7 "SOMBREA" "S" "0.15,0" "-0.075,-0.075" "0.075,0.075"
                                   "0,-0.055" "0,0.06" ""
           "UY" "DESIGNA" "P" "LT" ""
           "BORRA" "C" "-0.2,0.6" "0.2,-0.6" "E" "P" ""
           "DESIGNA" "LT" ""))

(defun PUNT () (setq P (list X Y) PX (mapcar '- P TG) PY (mapcar '+ P TG)))

(defun COMMUTA (LL)
  (subst (subst (if V (list X Y) N) (if V N (list X Y)) (nth Y LL))
         (nth Y LL) LL))

(defun V+- (C)
  (if V
      (command "BORRA" (ssname SS 0) "")
      (command "COPIA" SS "" "0,0" ""
               "CAMBIA" "LT" "" "P" "C" C "L" 1 "" "REGEN")))

(defun Y1 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 0) N1) ((= Y 3) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (+ Y K)
              (if (or (= N K3) (= N K4))
                  (+ Y (* 2 K))
                  Y))))

(defun Y2 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 1) N1) ((= Y 4) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (- Y K)
              (if (or (= N K3) (= N K4))
                  (+ Y K)
                  Y))))

(defun Y3 (K K1 K2 K3 K4)
  (setq N (if (= K 3) N4 (cond ((= Y 2) N1) ((= Y 5) N2) (T N3)))
        Y (if (or (= N K1) (= N K2))
              (- Y K)
              (if (or (= N K3) (= N K4))
                  (- Y (* 2 K))
                  Y))))

(defun V->N (/ N)
  (cond ((< Y 3) (Y1 3 2 3 4 5))
```

```
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 4 1 3))
        (T (Y3 3 1 4 3 5)))
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 3 4 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 4 1 3))
        (T (Y3 1 1 4 3 5)))
  (PUNT))

(defun N->V (/ N)
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 4 3 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 3 1 4))
        (T (Y3 1 1 3 4 5)))
  (cond ((< Y 3) (Y1 3 2 4 3 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 3 1 4))
        (T (Y3 3 1 3 4 5)))
  (PUNT))

(defun TV->TN (N1 N2 N3 N4 / N)
  (cond ((< Y 3) (Y1 3 2 3 4 5))
        ((and (< 2 Y) (< Y 6)) (Y2 3 2 4 1 3))
        (T (Y3 3 1 4 3 5)))
  (cond ((or (= Y 0) (= Y 3) (= Y 6)) (Y1 1 2 3 4 5))
        ((or (= Y 1) (= Y 4) (= Y 7)) (Y2 1 2 4 1 3))
        (T (Y3 1 1 4 3 5))))

(defun CLIC ()
  (if (> (getvar "CVPORT") 2)
      (progn
        (if (and (> 1-3 "1") Q) (setq P Q X (car P) Y (cadr P) PX QX PY QY))
        (setq N (ELEMENT **9**9**)
              V (atom (ELEMENT **V**V**))
              **V**V** (if (and (= 1-3 "1") (not V)) **V**V** (COMMUTA **V**V**))
              SS (ssget "_C" PX PY))
        (if (and (> 1-3 "1") (not Q)) (setq Q P QX PX QY PY))
        (V+- "VAR-1")
        (if (> 1-3 "1")
            (progn
              (V->N)
              (setq **V*V** (COMMUTA **V*V**))
              (if (= 1-3 "2")
                  (progn
                    (command "CVPORT" 2)
                    (setq SS (ssget "_C" PX PY))
                    (V+- "NORM-1")
                    (command "CVPORT" 3))))))
      (progn
        (setq N (ELEMENT **9*9**)
              V (atom (ELEMENT **V*V**))
              **V*V** (COMMUTA **V*V**)
              SS (ssget "_C" PX PY) Q P QX PX QY PY)
        (V+- "NORM-1")
        (command "CVPORT" 3)
        (N->V)
        (setq **V**V** (COMMUTA **V**V**)
              SS (ssget "_C" PX PY) R P RX PX RY PY)
        (V+- "VAR-1")
        (command "CVPORT" 2)
        (setq P Q X (car P) Y (cadr P) PX QX PY QY Q ())))
  (if (and (= 1-3 "2") (not FOTO2))
      (progn
        (setq TP (reverse (if (> (getvar "CVPORT") 2) Q R))
              X (car TP) Y (cadr TP))
        (TV->TN TN1 TN2 TN3 TN4)
        (setq T**V*V** (COMMUTA T**V*V**)
              TP (list X Y)
              VTV (if TRANSP T**V*V** **V*V**)
              P (if TRANSP TP P)))
      (if (> 1-3 "1") (setq VTV **V*V**))))
```

```
(defun FES-SUDOKU-1 ()
  (if (equal GR '(2 8))
      (setq PX (caar PPM) PY (cadar PPM)
            X (fix (/ (+ (car PX) (car PY)) 2))
            Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
            P (list X Y)))
  (CLIC)
  (if V
      (progn
        (if (= (last (car PPM)) 0)
            (while (< (last (car PPM)) 1)
                   (setq PPM (cdr PPM)
                         LLL/LLL (cdr LLL/LLL)
                         LLL (car LLL/LLL)
                         PX (caar PPM) PY (cadar PPM)
                         X (fix (/ (+ (car PX) (car PY)) 2))
                         Y (fix (/ (+ (cadr PX) (cadr PY)) 2))
                         N (ELEMENT **9**9**)
                         **V**V** (COMMUTA **V**V**)
                         SS (ssget "_C" PX PY))
                   (V+- "VAR-1")))
        (setq PPM (cdr PPM) LLL/LLL (cdr LLL/LLL) LLL (car LLL/LLL)))
      (progn
        (if (not LLL) (INI-LLL L1A9))
        (setq LLL (ACTUALITZA **V**V** N P LLL T) **V**V** (car LLL)
              LLL (cadr LLL) LLL/LLL (cons LLL LLL/LLL) GR N)
        (MASCARA (list N))
        (if (> M 0) (setq N GR PPM (cons (list PX PY M) PPM))))))

(defun NUMCOMPAT (P N NOU / PS)
  (setq K 0 KK 0)
  (foreach SUD SUDOKUS-V*V
    (setq K (1+ K)
          OK (if NOU (= (nth (car P) (nth (cadr P) SUD)) N) T)
          PS (if NOU (cons OK PS)))
    (if OK (progn
             (foreach S S-V*V (if (and OK (not (nth K S))) (setq OK ())))
             (if OK (setq KK (1+ KK))))))
  (if NOU (setq S-V*V (cons (cons P (reverse PS)) S-V*V))))

(defun RETOL-234 ()
  (RETOL-2 "Determinar" "quantes" "solucions" "compatibles"
           "hi ha" "pot trigar" "una bona" "estona." "Espereu..."))

(defun SUD-MIN (/ J0 J1 J2 J3 NOPOST)
  (setq I 0 J 0 K 0 POST T)
  (foreach LL (list **V*V** (TRANSPOSAR **V*V**))
    (if (and J1 POST)
        (if (= 1-3 "2")
            (if (< I 17) (setq POST ()))
            (progn
              (if (> I 16)
                  (setq PPM (cons (list (list P PX PY)
                                        (if (< (getvar "CVPORT") 3)
                                            (list R RX RY)
                                            (list Q QX QY)))
                                  PPM)))
              (if (< I NUM) (setq NOPOST T)))))
    (setq J3 1)
    (if POST
        (foreach L LL
          (if POST
              (progn
                (foreach E L
                  (if (atom E)
                      (progn
```

```
                        (if (or (and (not J1) (= J3 1))
                                (and J1 (not J2) (> J3 1)))
                            (setq I (1+ I)))
                        (setq J (1+ J)))))
                (setq J0 (cons J J0))
                (if (< K 2)
                    (setq K (1+ K))
                    (if (or (< J 2)
                            (equal J0 (list J 0 0))
                            (equal J0 (list J J 0))
                            (equal J0 (list J J J)))
                        (setq POST ())
                        (progn
                          (if J1
                              (if J2
                                  (setq J2 (if (< J J2) J J2)
                                        POST (or (< J3 3) (>= (+ J1 J2) 7)))
                                  (if (= J3 1)
                                      (setq J2 J)
                                      (setq J1 (if (< J J1) J J1))))
                              (setq J1 J))
                          (setq J0 () J 0 K 0 J3 (1+ J3)))))))))
  (if (= 1-3 "2")
      (if (and POST (= I 17) (= (min J1 J2) 2)) (setq POST ()))
      (if (and POST NOPOST) (setq POST ()))))

(defun C:CARREGA-123456789 (/ A)
  (setq RUTA (findfile "123456789.txt") A (open RUTA "r") L1 ())
  (repeat 362880 (setq L1 (cons (read-line A) L1)))
  (setq L1 (reverse L1))
  (close A))

 (defun DET-TRANSP (/ N Ñ)
  (setq N (N-3*3 **V*V** ()) Ñ (TRANSPOSAR N)
        N (1+2+3 N) Ñ (1+2+3 Ñ)
        TRANSP (> (- (eval (cons 'max N)) (eval (cons 'min N)))
                  (- (eval (cons 'max Ñ)) (eval (cons 'min Ñ))))
        VTV (if TRANSP T**V*V** **V*V**) P (if TRANSP TP P)))

(defun PERFIL-V*V (/ A B C V W L1C L2C LINIA-V*V KLINIA-V*V KKE KE)
  (if (not L1) (C:CARREGA-123456789))
  (DET-TRANSP)
  (setq FOTO1 (FOTO-V*V) *KK* () J 8 K -1)
  (repeat 3
    (setq J (- J 9) K (+ K 3))
    (repeat 3
      (setq J (+ J 3) K (- K 3) W ())
      (repeat 3
        (setq J (- J 3) K (1+ K))
        (repeat 3
          (setq J (1+ J) V (nth J (nth K VTV)))
          (if (atom V) (setq W (cons V W)))))
      (cond ((= J 2) (setq A W))
            ((= J 5) (setq B W))
            (T       (setq C W))))
    (repeat 3 (setq L1C (cons (list A A A B B B C C C) L1C))))
  (setq L1C (reverse L1C) J -1)
  (repeat 9
    (setq J (1+ J) K -1 L ())
    (repeat 9 (setq K (1+ K) C (nth J (nth K VTV))
                    L (if (atom C) (cons C L) L)))
    (setq L2C (cons L L2C)))
  (setq L2C (reverse L2C) LINIES-V*V () KLINIES-V*V () K -1)
  (terpri)
  (repeat 9
    (setq LINIA-V*V () KLINIA-V*V () L "" K (1+ K))
    (mapcar '(lambda (E F G)
               (setq L (strcat L (if (or F G)
```

```
                                        (if (atom E)
                                            (itoa E)
                                            (progn
                                              (setq C "]")
                                              (foreach H (append F G)
                                                (setq C (strcat (itoa H) C)))
                                              (strcat "[~" C)))
                                        "#"))))
            (nth K VTV) (nth K L1C) L2C)
    (cond ((= K 0) (setq A "123456789" B "198765432"))
          ((= K 1) (setq A "213456789" B "897654321"))
          ((= K 2) (setq A "312456789" B "987654321"))
          ((= K 3) (setq A "213456789" B "498765321"))
          ((= K 4) (setq A "312456789" B "897654321"))
          ((= K 5) (setq A "412356789" B "987654321"))
          ((= K 6) (setq A "312456789" B "798654321"))
          ((= K 7) (setq A "412356789" B "897654321"))
          (T       (setq A "512346789" B "987654321")))
    (foreach E (reverse (member B (reverse (member A L1))))
      (if (wcmatch E L)
          (progn
            (setq LINIA-V*V (cons E LINIA-V*V) KKE () *K* 10)
            (repeat 3
              (setq KE ())
              (repeat 3 (setq *K* (1- *K*) KE (cons (substr E *K* 1) KE)))
              (setq KKE (cons KE KKE)))
            (setq KLINIA-V*V (cons KKE KLINIA-V*V)))))
    (setq LINIES-V*V (cons (reverse LINIA-V*V) LINIES-V*V)
          KLINIES-V*V (cons (reverse KLINIA-V*V) KLINIES-V*V)
          *K* (itoa (length LINIA-V*V)) *KK* (cons *K* *KK*))
    (terpri)
    (prompt (strcat "\nEntre 362880 línies, de compatibles amb la " (itoa (1+ K))
                    "ª de la V. E. NORM. n'hi ha " *K* "."))
  (setq LINIES-V*V (reverse LINIES-V*V) KLINIES-V*V (reverse KLINIES-V*V)
        L1 () *KK* (reverse *KK*))
  (terpri))

(defun KOMPAT-2L ()
  (setq OK T)
  (mapcar '(lambda (L0 L1)
             (foreach E L1 (if (and OK (member E L0)) (setq OK ()))))
          KT0 KT1)
  (if OK (setq KT01 (list (append (cadr KT0) (cadr KT1))
                          (append (last KT0) (last KT1))))))

(defun KOMPAT-3L ()
  (setq OK T)
  (mapcar '(lambda (L01 L2)
             (foreach E L2 (if (and OK (member E L01)) (setq OK ()))))
          KT01 (cdr KT2))
  OK)

(defun ACTLOC () (setq LOC (cons (cons A K) LOC)))

(defun LOCT (TT / A Z LOC L1)
  (setq A (substr (caar TT) 1 1) Z (substr (car (last TT)) 1 1) K 0)
  (ACTLOC)
  (while (< A Z)
         (setq A (itoa (1+ (atoi A))))
         (while (< (car (nth (setq K (1+ K)) TT)) A))
         (setq A (substr (car (nth K TT)) 1 1))
         (ACTLOC))
  (reverse LOC))

(defun COLS-3L (I / C CC)
  (setq K 10)
  (repeat 9
    (setq K (1- K) C "")
```

```
    (foreach J I (setq C (strcat C (substr J K 1))))
    (setq CC (cons C CC))))

(defun SUD-NUM (/ LN LLN)
  (foreach 3L (list T2 T1 T0)
    (foreach L (reverse 3L)
      (setq LN () K 10)
      (repeat 9 (setq K (1- K) LN (cons (atoi (substr L K 1)) LN)))
      (setq LLN (cons LN LLN)))))

(defun MEMBRE (K L) (repeat K (setq L (cdr L))) L)

(defun ACLARIMENT ()
  (terpri) (***)
  (prompt "\n Per tal d'abreujar aquest procés, s'ha substituït la VERSIÓ")
  (prompt " ESCOLLIDA per la\n seva transposada (substitució de files per")
  (prompt " columnes), però com que seria poc\n entenedor al•ludir a \"")
  (prompt "solucions normalitzades compatibles amb el resultat de\n normalitzar")
  (prompt " la transposició de la VERSIÓ ESCOLLIDA\", no s'ha canviat el text.")
  (***)
  (prompt "\nA partir d'ara, l'expressió \"solucions normalitzades\"")
  (prompt " s'haurà d'entendre així.")
  (***) (terpri))

(defun COMPAT-PERFILS (/ L1 LL0 KLL0 LL1 KLL1 L2 LL2 KLL2 KT01 3L M N TERCETS-V*V
                         CC0 PCC0 CC1 PCC01 CC2 CCC1 CCC2 TT0 TT1 TT2 LOCT1 LOCT2
                         TOT *T* *TT*)
  (setq SUDOKUS-V*V () S-V*V () K 0 *K* 0)
  (repeat 9 (setq K (1+ K) TOT (cons (strcat "*" (itoa K) "*") TOT)))
  (repeat 3
    (setq LL0 (car LINIES-V*V) LL1 (cadr LINIES-V*V) LL2 (caddr LINIES-V*V)
          LINIES-V*V (cdddr LINIES-V*V) TERCETS-V*V ()
          KLL0 (car KLINIES-V*V) KLL1 (cadr KLINIES-V*V) KLL2 (caddr KLINIES-V*V)
          KLINIES-V*V (cdddr KLINIES-V*V))
    (mapcar '(lambda (T0 KT0)
               (mapcar '(lambda (T1 KT1)
                          (if (and (< T0 T1) (KOMPAT-2L))
                              (mapcar '(lambda (T2 KT2)
                                         (if (and (< T1 T2) (KOMPAT-3L))
                                             (setq TERCETS-V*V
                                                   (cons (list T0 T1 T2)
                                                         TERCETS-V*V))))
                                      LL2 KLL2)))
                        LL1 KLL1))
             LL0 KLL0)
    (if TERCETS-V*V (setq *T* (itoa (length TERCETS-V*V))
                          *TT* (append *TT* (list *T*))
                          SUDOKUS-V*V (append SUDOKUS-V*V
                                              (list (reverse TERCETS-V*V)))))
    (prompt (strcat "\nEntre " (nth *K* *KK*) " línies compatibles amb la "
                    (itoa (setq *K* (1+ *K*))) "ª, " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª i " (nth *K* *KK*) " amb la "
                    (itoa (setq *K* (1+ *K*))) "ª,\nhi ha " *T* " tríades "
                    "normalitzades compatibles amb la V. E. NORMALITZADA.\n")))
  (if (> (length SUDOKUS-V*V) 2)
      (progn
        (setq KLINIES-V*V () TERCETS-V*V ()
              TT0 (car SUDOKUS-V*V) TT1 (cadr SUDOKUS-V*V) TT2 (last SUDOKUS-V*V)
              SUDOKUS-V*V () TOT () LOCT1 (LOCT TT1) LOCT2 (LOCT TT2) K 10)
        (repeat 8 (setq K (1- K) TOT (cons (itoa K) TOT)))
        (foreach T1 (reverse TT1) (setq CC1 (COLS-3L T1) CCC1 (cons CC1 CCC1)))
        (foreach T2 (reverse TT2) (setq CC2 (COLS-3L T2) CCC2 (cons CC2 CCC2)))
        (foreach T0 TT0
          (if (setq CC0 (COLS-3L T0)
                    PCC0 (mapcar '(lambda (C0) (strcat "*[" C0 "]*")) CC0)
                    L1 (SUPR (substr (cadr T0) 1 1) TOT)
                    L1 (SUPR (substr (last T0) 1 1) L1)
                    PRIM1 (cdr (assoc (car L1) LOCT1)))
```

```
            (mapcar '(lambda (T1 CC1)
                       (setq L2 (cdr L1) K -1 OK T)
                       (while (and OK (< (setq K (1+ K)) 9))
                              (if (wcmatch (nth K CC1) (nth K PCC0))
                                  (setq OK ())))
                       (if (and OK (setq PCC01
                                         (mapcar '(lambda (C0 C1)
                                                    (strcat "*[" C0 C1 "]*"))
                                                 CC0 CC1)
                                         L2 (SUPR (substr (cadr T1) 1 1) L2)
                                         L2 (SUPR (substr (last T1) 1 1) L2)
                                         PRIM2 (cdr (assoc (car L2) LOCT2))))
                           (mapcar '(lambda (T2 CC2)
                                      (setq K -1 OK T)
                                      (while (and OK (< (setq K (1+ K)) 9))
                                             (if (wcmatch (nth K CC2)
                                                          (nth K PCC01))
                                                 (setq OK ())))
                                      (if OK (setq SUDOKUS-V*V
                                                   (cons (SUD-NUM) SUDOKUS-V*V)
                                                   S-V*V (cons T S-V*V))))
                                   (member (nth PRIM2 TT2) TT2)
                                   (MEMBRE PRIM2 CCC2))))
                     (member (nth PRIM1 TT1) TT1)
                     (MEMBRE PRIM1 CCC1))))
        (setq TT0 () TT1 () TT2 () CCC1 () CCC2 ()
              KK (length S-V*V) S-V*V (list (cons P S-V*V)))
        (prompt (strcat "\nEntre les " (nth 0 *TT*) " tríades 1-2-3, les "
                        (nth 1 *TT*) " tríades 4-5-6 i les " (nth 2 *TT*)
                        " tríades 7-8-9,\nde solucions normalitzades compatibles "
                        "amb la V. E. NORMALITZADA n'hi ha " (itoa KK) "."))
        (textscr)
        (if TRANSP (ACLARIMENT))
        (SEGUIR ())
        (graphscr))))

(defun RE+MEMBER (R-9*9 R-LLL / 2R R-N R-P F C0 C3 C6)
  (if (or 1-3<34 (< (length KKK) 2))
      (if (COMPLET-9*9 R-9*9)
          (setq SUDOKUS-V*V (cons (if NORM=1 (TRANSPOSAR R-9*9) R-9*9)
                                  SUDOKUS-V*V)
                KKK (cons T KKK))
          (progn
            (if (and PRE (> (cadr R-P) 0))
                (setq PRE () F (nth 0 R-9*9)
                      C0 (nth 0 F) C3 (nth 3 F) C6 (nth 6 F)
                      NORM (and (< C0 C3 C6)
                                (< C0 (nth 1 F) (nth 2 F))
                                (< C3 (nth 4 F) (nth 5 F))
                                (< C6 (nth 7 F) (nth 8 F)))))
            (if (or PRE (not NORM))
                (progn
                   (setq R-N (nth (car R-P) (nth (cadr R-P) R-LLL)))
                  (foreach E R-N
                    (if E (progn
                            (setq 2R (ACTUALITZA R-9*9 E R-P R-LLL ()))
                            (RE+MEMBER (car 2R) (cadr 2R))))))
                (if NORM (setq PRE T)))))))

(defun ANORMALS (NORM=1 1-3<34 / L0A8 L LL LLL V X XX XXX Y YY YYY XY R-P PRE
                                 NORM)
  (if TRANSP (setq VTV **V*V** TRANSP ()))
  (setq L0A8 '(0 1 2 3 4 5 6 7 8) FOTO1 () FOTO2 (FOTO-V*V))
  (if (not NORM=1) (setq KKK () SUDOKUS-V*V ()))
  (foreach X L0A8
    (setq XX ())
    (foreach Y L0A8 (if (atom (setq V (ELEMENT **V*V**))) (setq XX (cons V XX))))
    (setq XXX (cons XX XXX)))
```

```
  (setq XXX (reverse XXX))
  (foreach Y L0A8
    (setq YY ())
    (foreach X L0A8 (if (atom (setq V (ELEMENT **V*V**))) (setq YY (cons V YY))))
    (setq YYY (cons YY YYY)))
  (setq YYY (reverse YYY))
  (foreach YY '(0 3 6)
    (foreach XX '(0 3 6)
      (setq L () Y -1)
      (foreach VV **V*V**
        (setq Y (1+ Y) X -1)
        (if (not (or (< Y YY) (> Y (+ YY 2))))
            (progn
              (foreach V VV
                (setq X (1+ X))
                (if (not (or (< X XX) (> X (+ XX 2))))
                    (if (atom V) (setq L (cons V L))))))))
      (setq XY (cons L XY))))
  (setq XY (reverse XY))
  (foreach Y L0A8
    (setq LL ())
    (foreach X L0A8
      (setq L ())
      (if (listp (ELEMENT **V*V**))
          (foreach E L1A9
            (setq L (cons (if (or (member E (nth X XXX)) (member E (nth Y YYY))
                                  (member E (nth (+ (/ X 3) (* (/ Y 3) 3)) XY)))
                              ()
                              E)
                          L))))
      (setq LL (cons (reverse L) LL)))
    (setq LL (reverse LL) LLL (cons LL LLL)))
  (setq LLL (reverse LLL))
  (if NORM=1
      (progn
        (setq PRE T LL ())
        (foreach EE (TRANSPOSAR **V*V**)
          (setq L ())
          (foreach E EE (setq L (cons (if (atom E) E (reverse E)) L)))
          (setq LL (cons (reverse L) LL)))
        (setq LL (reverse LL))
        (RE+MEMBER LL (TRANSPOSAR LLL)))
      (RE+MEMBER **V*V** LLL))
  (setq KK (length KKK) KKK (cons P KKK) S-V*V (list KKK))
  (if (and NORM=1 (= KK 1)) (setq KKK ())))

(defun FES-SUDOKU-234 (/ LINIES-V*V KLINIES-V*V *K* *KK* R RX RY)
  (setq Q ())
  (CLIC)
  (if POST
      (if V
          (if (or (member P FOTO1) (member P FOTO2))
              (if (= 1-3 "4")
                  (setq POST ())
                  (progn
                    (command "ESPACIOP" "BORRA" "LT" ""
                             "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                    (RETOL-2 "Anul-lant" "aquesta" "casella," "caldrà"
                             "recalcular" "solucions" "compatibles"
                             "i tornar a" "esperar...")
                    (if FOTO2
                        (progn
                          (SEGUIR T)
                          (command "DESHACER" "R" "DESHACER" "M"
                                   "INSERT" "C" "0,0" "" "" "")
                          (RETOL-2 "...i, si surt" "\"EL SUDOKU" "JA TÉ UNA"
                                   "SOLUCIÓ" "ÚNICA\", mai" "no sabreu"
                                   "si sobren" "caselles" "plenes.")))
```

```
                    (initget "Si No")
                    (setq CONF
                          (getkword "\n.\n.\nConfirmeu l'anul·lació (S/N)? <N>: ")
                          CONF (if CONF CONF "No"))
                    (if (= CONF "Si")
                        (setq POST ())
                        (progn
                          (command "DESHACER" "R" "UY" "ESPACIOM")
                          (CLIC)))))
              (progn
                (setq I (assoc P S-V*V) S-V*V (SUPR I S-V*V)
                      I (caar S-V*V) J (nth (car I) (nth (cadr I) VTV)))
                (NUMCOMPAT I J ())))
            (progn
              (command "ESPACIOP" "DESHACER" "M"
                       "INSERT" "C" "0,0" "" "" "")
              (RETOL-234)
              (NUMCOMPAT P N T)
              (command "DESHACER" "R"))))
(if (not POST)
    (progn
      (if (and V (or (member P FOTO1) (member P FOTO2)))
          (setq POST T)
          (if (not V) (SUD-MIN)))
      (if POST
          (progn
            (if V
                (if (< 1-3 "4") (command "DESHACER" "R") (command "ESPACIOP"))
                (progn
                  (command "ESPACIOP")
                  (if (= 1-3 "2") (command "BORRA" "LT" ""))))
            (if (or (not (or FOTO1 FOTO2))
                    (member P FOTO1))
                (if (= 1-3 "2")
                    (progn
                      (command "DESHACER" "M" "INSERT" "C" "0,0" "" "" "")
                      (RETOL-2 "Determinar" "quantes" "solucions" "normalitza-"
                               "des són" "compatibles" "pot trigar" "hores."
                               "Espereu...")
                      (PERFIL-V*V)
                      (COMPAT-PERFILS)
                      (command "DESHACER" "R" "UY" "ESPACIOM"))
                    (progn
                      (prompt "\n.\n.\n")
                      (command "INSERT" "C" "0,0" "" "" ""
                               "DESHACER" "M")
                      (RETOL-234)
                      (ANORMALS () T)
                      (command "DESHACER" "R")
                      (if (= KK 1)
                          (progn
                            (while
                              (and (= KK 1) (> (length PPM) 1))
                              (prompt "Només hi ha una solució compatible i, ")
                              (prompt "just per això, cal mirar\nquantes n'hi")
                              (prompt " hauria amb una casella menys (")
                              (prompt (itoa (setq NUM (1- NUM))))
                              (prompt " caselles plenes).")
                              (SEGUIR ())
                              (command "ESPACIOM")
                              (setq Q (car PPM) PPM (cdr PPM)
                                    Q (if (< (getvar "CVPORT") 3)
                                          (car Q)
                                          (cadr Q))
                                    P (car Q) X (car P) Y (cadr P)
                                    PX (cadr Q) PY (last Q) Q ())
                              (CLIC)
                              (command "ESPACIOP")
```

```
                                (if (> (length PPM) 0)
                                    (progn
                                      (prompt "\n.\n.\n")
                                      (command "DESHACER" "M")
                                      (RETOL-234)
                                      (ANORMALS () ())
                                      (command "DESHACER" "R"))))
                                (if (> KK 1)
                                    (progn
                                      (prompt "Us heu passat de la ratlla: ")
                                      (prompt "amb més d'una solució compatible, ")
                                      (prompt "el sudoku\nhaurà de recuperar ")
                                      (prompt "l'última casella esborrada (")
                                      (prompt (itoa (setq NUM (1+ NUM))))
                                      (prompt " caselles plenes).")
                                      (SEGUIR ())
                                      (command "ESPACIOM")
                                      (CLIC)
                                      (command "ESPACIOP")
                                      (NUMCOMPAT P N T))
                                    (prompt (strcat "\n.\nHeu tingut el privilegi "
                                                    "de crear un SUDOKU MÍNIM "
                                                    "(només 17 caselles plenes)."))))))
                            (command "BORRA" "LT" "")))
                      (progn
                        (command "DESHACER" "M"
                                 "INSERT" "C" "0,0" "" "" "")
                        (RETOL-234)
                        (ANORMALS () (< 1-3 "4"))
                        (command "DESHACER" "R")
                        (if (< 1-3 "4") (command "UY" "ESPACIOM")))))))))
(if POST
    (progn
      (setq GR (strcat "\nEL SUDOKU JA TÉ UNA SOLUCIÓ ÚNICA.\nOmplint més case"
                       "lles el fareu més fàcil (<Intro> o < > per acabar)"))
      (if (or (and (= 1-3 "4") (= KK 1) (> (setq NUM (length FOTO2)) 17))
              (and (< 1-3 "4") (> KK 1)))
          (if (= 1-3 "4")
              (prompt (strcat "\n.\nAmb només una solució compatible, heu de "
                              "buidar més caselles:"))
              (prompt (strcat "\n.\nAmb " (itoa KK) " solucions "
                              (if KKK "" "normalitzades ")
                              "heu d'omplir més caselles:")))
          (if FOTO2
              (progn
                (if (= 1-3 "4")
                    (progn
                      (if (> KK 1)
                          (progn
                            (command "ESPACIOP" "DESHACER" "M"
                                     "INSERT" "C" "0,0" "" "" "")
                            (prompt "\n.\nUs heu passat de la ratlla: amb més ")
                            (prompt "d'una solució compatible, el sudoku\n")
                            (prompt "haurà de recuperar l'última casella ")
                            (prompt (strcat "esborrada (" (itoa NUM)))
                            (prompt " caselles plenes).")
                            (SEGUIR ())
                            (command "DESHACER" "R" "ESPACIOM")
                            (CLIC)
                            (command "ESPACIOP" "DESHACER" "M"
                                     "INSERT" "C" "0,0" "" "" "")
                            (NUMCOMPAT P N T)
                            (command "DESHACER" "R")))
                      (setq 1-3 "3")))
                (if KKK (setq FOTO2 (FOTO-V*V) KKK ()))
                (prompt (strcat (if (and (> 1-3 "2")
                                         (or (= NUM 17)
                                             (= (length (FOTO-V*V)) 17)))
```

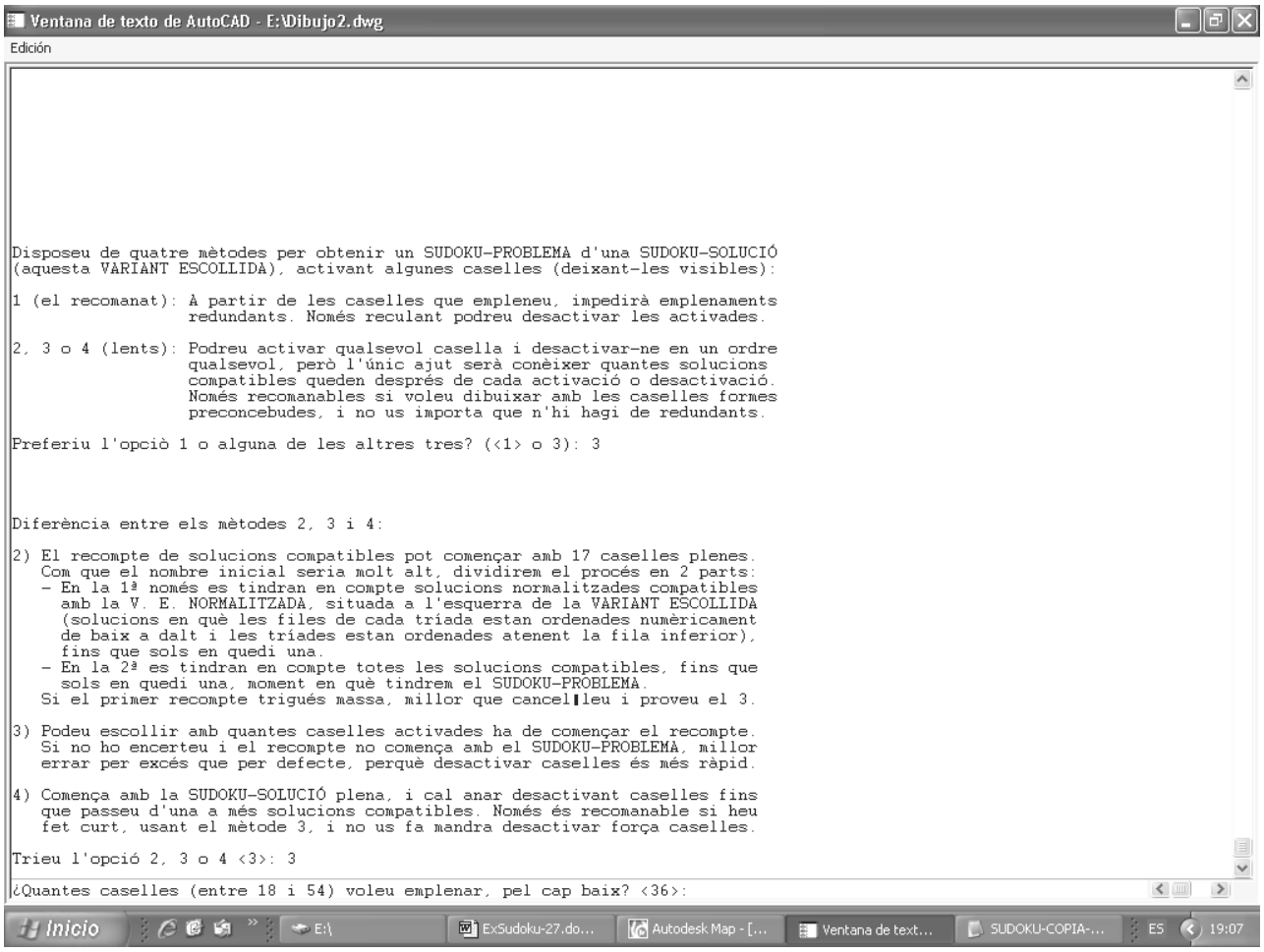
```
                                    (progn
                                      (setq PPM () NUM ())
                                      (strcat "\n.\nHeu tingut el privilegi de "
                                              "crear un SUDOKU MÍNIM (només 17 "
                                              "caselles plenes)."))
                                    "\n.")
                                  "\n" GR)))
                (progn
                  (foreach E '(T ())
                    (prompt (strcat "\n.\nTeniu 1 solució normalitzada compatible"
                                    " amb V. E. NORMALITZADA."))
                    (if E (SEGUIR ()) (terpri)))
                  (command "ESPACIOP"
                           "DESHACER" "M"
                           "INSERT" "C" "0,0" "" "" "")
                  (RETOL-2 "Però pot" "haver-n'hi" "més, de no" "normalitza-"
                           "des però" "igualment" "compatibles." "Així que"
                           "espereu...")
                  (ANORMALS T T)
                  (command "DESHACER" "R" "UY" "ESPACIOM")
                  (if (> KK 1)
                      (progn
                        (if (> KK 2)
                            (prompt (strcat "\n.\nHi ha " (itoa (1- KK))
                                            " solucions més, no normalitzades"
                                            " però també compatibles,"))
                            (prompt (strcat "\n.\nHi ha una solució més, no nor"
                                            "malitzada però també compatible,")))
                        (prompt "\naixí que heu d'omplir més caselles:"))
                      (prompt (strcat "\n.\nCom no n'hi ha cap de "
                                      "no normalitzada, " GR)))))))
  (if (= (getvar "CVPORT") 1) (command "ESPACIOM")))

(defun FES-SUDOKU (/ KK KKK L1 RUTA SUDOKUS-V*V S-V*V FOTO1 FOTO2 LLL/LLL **V*V**
                     T**V*V** **V**V** VTV TRANSP T**9*9**)
  (if (= 1-3 "1")
      (progn
        (INI-LLL ())
        (setq NIL*NIL LLL 0*0 (INI-COORDS) POST T)
        (INI-LLL L1A9))
      (progn
        (repeat 2 (VAR->NORM) (PRE-NORM->VAR))
        (if (= 1-3 "2") (EXPLICACIO))))
  (setq PPM () **V**V** (if (< 1-3 "4") (INI-COORDS) **9**9**)
        **V*V** (if (> 1-3 "1") (if (< 1-3 "4") **V**V** **9*9**)) K 1
        GR (if (= 1-3 "1")
               (strcat "Sobre la VARIANT ESCOLLIDA (dreta), feu clic en les "
                       "caselles que hagi de veure\nel jugador (per anul•lar "
                       "els últims clics, polseu la tecla \"<---Backspace\"):")
               (strcat "Podeu fer clic a qualsevol casella, sobre la "
                       (if (= 1-3 "2")
                           (strcat "finestra dreta o sobre l'esquerra\n(es "
                                   "passa d'una a l'altra amb un clic previ):")
                           "VARIANT ESCOLLIDA (dreta):"))))
  (prompt (strcat "\n.\n.\n" GR))
  (if (= 1-3 "2")
      (progn
        (foreach EE (reverse (TRANSPOSAR **V*V**))
          (setq J ())
          (foreach E EE (setq J (cons (reverse E) J)))
          (setq T**V*V** (cons (reverse J) T**V*V**))))
      (if (= 1-3 "4") (setq VTV **V*V** FOTO2 (FOTO-V*V) POST T)))
  (repeat 2
    (setq K (1+ K) J (if (= K 2) "NORM-0" "VAR-0"))
    (if (or (> K 2) (= 1-3 "2") (> ABC "A"))
        (progn
          (command "ESPACIOM" "CVPORT" K)
```

```
          (if (and (> K 2) (= 1-3 "4"))
              (command "COPIA" "C" "0,0" "8,8" "" "0,0" ""
                       "CAMBIA" "P" "" "P" "L" 1 ""
                       "BORRA" "P" ""))
          (command "CAMBIA" "C" "0,0" "8,8" "" "P" "C" J "")
          (if (and (> K 2) (= 1-3 "4"))
              (command "UY" "ORDENAOBJETOS" "P" "" "D")))))
  (graphscr)
  (YIN-YANG)
  (command "ESPACIOM")
  (while (not (or (and (= 1-3 "1") (= M 0))
                  (setq GR (grread T)
                        FORA (or (equal GR '(2 13))
                                 (equal GR '(2 32)) (= (car GR) 25)))))
         (if (or (= 1-3 "2") (= (getvar "CVPORT") 3))
             (if (= (car GR) 5)
                 (command "ESPACIOP" "GIRA" "LT" "" "0,0" ANG "ESPACIOM")
                 (if (or (and (= 1-3 "1") PPM (equal GR '(2 8)))
                         (and (= (car GR) 3)
                              (> (getvar "CVPORT") 1)
                              (setq P (cadr GR) PX (car P) PY (cadr P))
                              (equal (list PX PY) '(4 4) 4.48)
                              (or (< (- PX (setq X (fix PX))) 0.48)
                                  (< (- (setq X (fix (1+ PX))) PX) 0.48))
                              (or (< (- PY (setq Y (fix PY))) 0.48)
                                  (< (- (setq Y (fix (1+ PY))) PY) 0.48))
                              (or (> 1-3 "1") (listp (ELEMENT **V**V**)))
                              (PUNT)))
                     (if (= 1-3 "1") (FES-SUDOKU-1) (FES-SUDOKU-234))))))
  (if (and FORA (= (substr (getvar "LASTPROMPT") 1 9) "EL SUDOKU"))
      (setq FORA ()))
  (if (not FORA)
      (progn
        (if (= 1-3 "1")
            (progn
              (command "CAPA" "DES" "VAR-0" "")
              (foreach E PPM
                (if (= (last E) 0) (command "BORRA" "C" (car E) (cadr E) "")))
              (command "CAPA" "ACT" "VAR-0" "")))
        (RASTRE)
        (command "ESPACIOP" "BORRA" "C" "-0.22,-0.53" "0.22,0.53"
                                    "C" "-0.75,-0.58" "0.75,-0.52" "")
        (RETOLS))))

(defun C:SUDOKULUM (/ ANG TG NUM L1A9 LLISTES CONF ABC 1-3 0*0 3*N NIL*NIL G I J
                      K L LL LLL M N N1 N2 N3 N4 P PPM PX PY Q QX QY SS V X Y OK
                      GR POST OSN FIL SRT SVT ECO TN/R-L TT **9*9** **9**9** FORA
;                     HPD DRA LAY                          ; Només si és ACAD 2011.
                    )
  (setq ANG -0.5   ; Un angle que giri el YIN-YANG a una velocitat proporcionada a
        TG '(0.48 0.48) L1A9 '(1 2 3 4 5 6 7 8 9)                ; la del cursor.
        NUM 36)  ; Valor per defecte en nombre de caselles plenes que activa POST
  (foreach N1 L1A9                                              ; (funció SUD-MIN).
    (foreach N2 (cdr (member N1 L1A9))
      (foreach N3 (cdr (member N2 L1A9)) (setq 3*N (cons (list N1 N2 N3) 3*N)))))
  (setq 3*N (reverse 3*N))
  (PREPGRAF-9*9)
  (while (and (/= CONF "Si") (/= ABC "D"))
         (repeat 40 (terpri))
         (if (= CONF "No")
             (progn
               (if (> (getvar "CVPORT") 1) (command "ESPACIOP"))
               (command "BORRA" "C" "-0.75,-0.58" "0.75,-0.52" "" "ESPACIOM"
                        "CVPORT" 2 "BORRA" "C" "0,0" "8,8" ""
                        "CVPORT" 3 "BORRA" "C" "0,0" "8,8" "")
               (***)
               (prompt "\n*****************************  NOVA SOLUCIÓ")
```

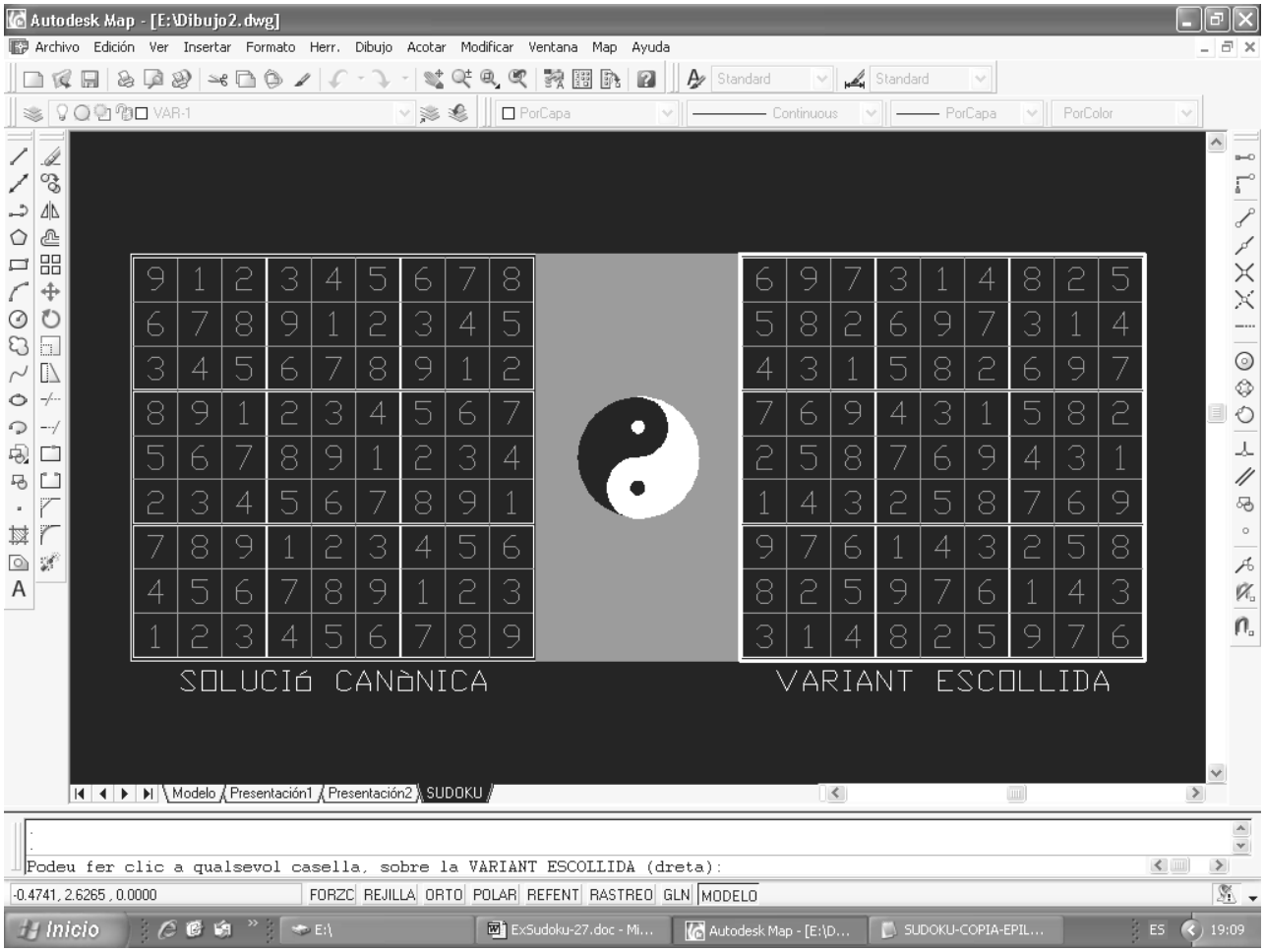
```
              (prompt "  ******************************")
              (***)
              (terpri)))
          (PREPTEXT-9*9)
          (if (= ABC "C")
              (setq **9*9** (list '(1 2 3 4 5 6 7 8 9) '(4 5 6 7 8 9 1 2 3)
                                  '(7 8 9 1 2 3 4 5 6)
                                  '(2 3 4 5 6 7 8 9 1) '(5 6 7 8 9 1 2 3 4)
                                  '(8 9 1 2 3 4 5 6 7)
                                  '(3 4 5 6 7 8 9 1 2) '(6 7 8 9 1 2 3 4 5)
                                  '(9 1 2 3 4 5 6 7 8)))
              (FES-SOLUCIO))
          (if (/= ABC "D")
              (progn
                (setq **9**9** **9*9**)
                (if (= ABC "C") (progn (command "ESPACIOM") (GRAF-9*9 ())))
                (command "ESPACIOP")
                (RETOL-1 '(-0.75 -0.55) (strcat "SOLUCIÓ "
                                                (cond ((= ABC "A") "CREADA")
                                                      ((= ABC "B") "COPIADA")
                                                      (T "CANÒNICA"))))
                (RETOL-1 '(0.75 -0.55) "VARIANT ESCOLLIDA")
                (SEGUIR T)
                (VARIACIONS)
                (CANON->VARIANT))))
    (if (/= ABC "D") (progn (TRIA-METODE) (FES-SUDOKU)))
    (if FORA
        (prompt "\n*** EXECUCIÓ INTERROMPUDA PER L'USUARI ***")
        (progn
          (prompt "\n.")
          (if (= 1-3 "1")
              (prompt "\nJa teniu un SUDOKU-PROBLEMA amb les caselles justes!"))))
    (prompt "\n(Als efectes de revocació, l'execució de SUDÒKULUM ")
    (prompt "compta com una sola ordre.)")
    (command "OSMODE" OSN
             "FILLMODE" FIL
             "SORTENTS" SRT
             "SAVETIME" SVT
;            "HPDRAWORDER" HPD                              ; Només si és ACAD 2011.
;            "DRAWORDERCTL" DRA                             ; Només si és ACAD 2011.
;            "LAYLOCKFADECTL" LAY                           ; Només si és ACAD 2011.
             "DESHACER" "F")
    (setvar "CMDECHO" ECO)
    (princ))
```

I res més, sinó acomiadar-nos del sofert lector, encoratjant-lo a fer tot allò que l'autor entén que convindria però que renuncia a conduir personalment, en haver-se esgotat el termini màxim de 5 anys que s'havia autoconcedit i sentir la imperiosa necessitat vital de canviar de tema. Bàsicament, aquestes tasques pendents serien:

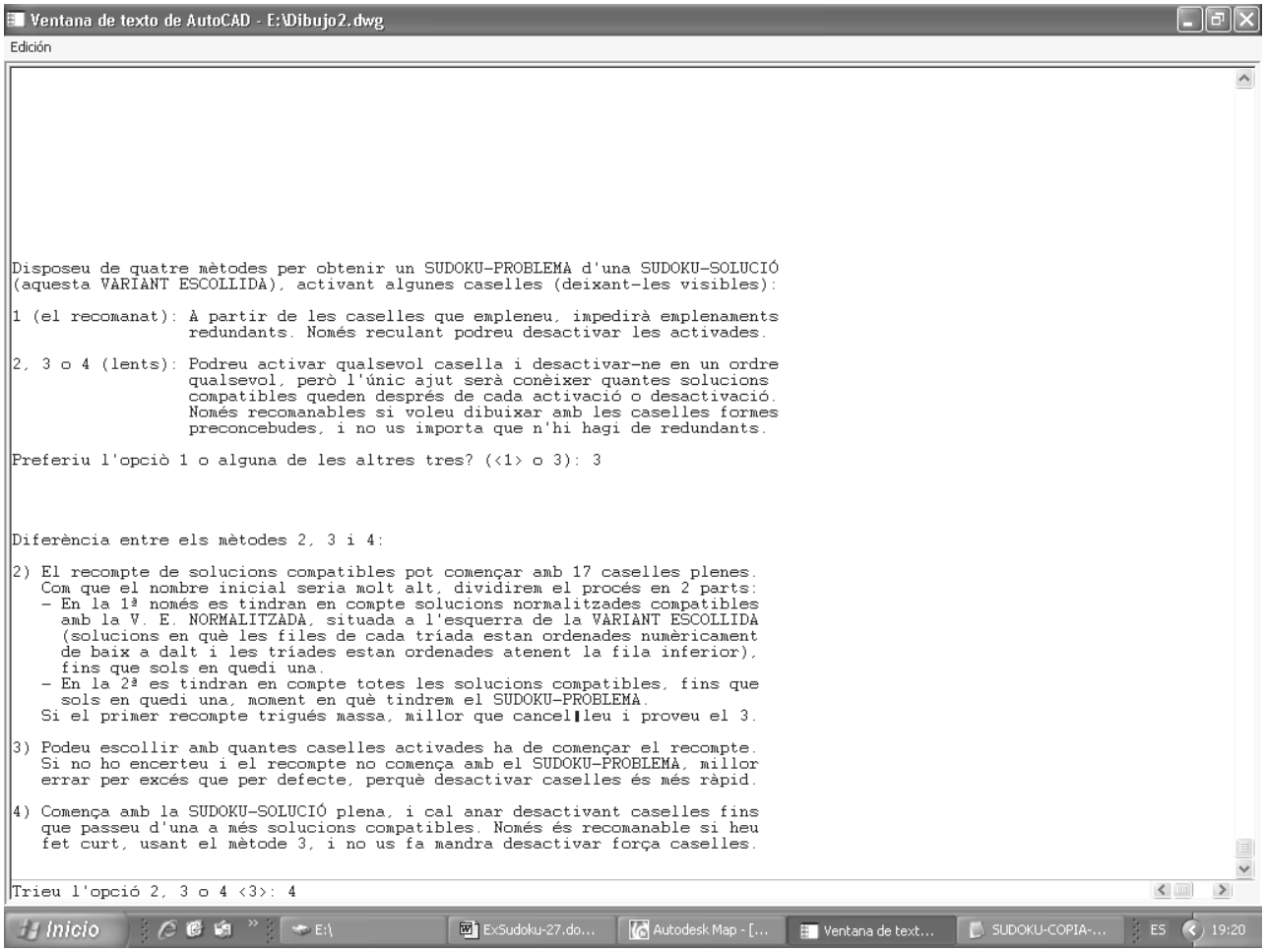
- Desempallegar-se del mètode 2, i tenir la paciència d'expurgar el codi bandejant del procés aquelles marrades que únicament es justificaven en els avantatges que reportava l'ús de solucions normalitzades en la primera etapa del mètode (recurs a ****9*9**** i a ****V*V****) i la discutible oportunitat que l'usuari pogués manipular directament aquestes solucions (activant/desactivant les caselles en la finestra V. E. NORMALITZADA).
- Integrar allò que pugui haver-hi d'aprofitable en el codi presentat al final del capítol 1 (*Inventariar les solucions: un cul-de-sac*); per exemple, transformant l'opció A, que en lloc de SOLUCIÓ CANÒNICA podríem anomenar SOLUCIÓ NORMALITZADA i que en comptes de limitar-se a una solució concreta passaria a representar la posssibilitat de carregar qualsevol de les solucions normalitzades inventariades en arxius del tipus XSN-<pF1>-<pF4>-<pF7> (les que haguem pogut crear, és clar, visualitzades mitjançant **C:VIS-XSN**), que "desnormalitzaríem" amb **CANON->VARIANT**, convertida en l'habitual VARIANT ESCOLLIDA. L'anomenada SOLUCIÓ CANÒNICA podria quedar com a solució per defecte, dins d'aquesta nova concepció d'opció A.
- Integrar en **VARIACIONS** i **CANON->VARIANT** una nova transformació: la que podríem anomenar "3×3" (la permutació en bloc de requadres 3×3, vista a la pàgina 116) i que, com F3 i C3, seria d'aplicació restringida a només algunes SUDOKUs-SOLUCIÓ.



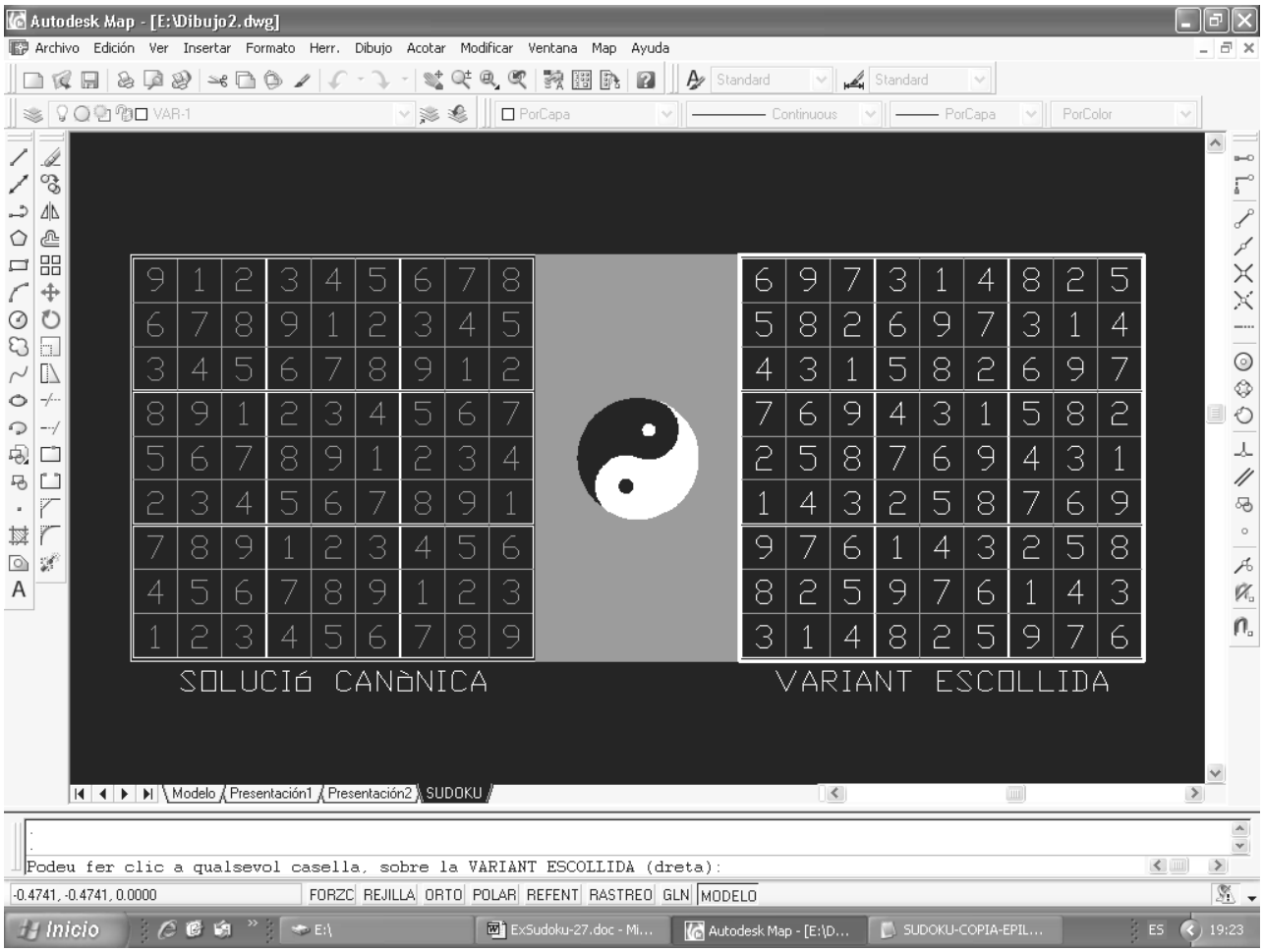
Com que, utilitzant els mètodes 3 i 4, la V. E, NORMALITZADA ja no hi intervé...



(almenys externament), la activació/desactivació de caselles queda restringida...



a la finestra dreta (VARIANT ESCOLLIDA). L'esquerra és una presència passiva i...



testimonial, que ens en recorda l'origen: la SOLUCIÓ CREADA, COPIADA o CANÒNICA.